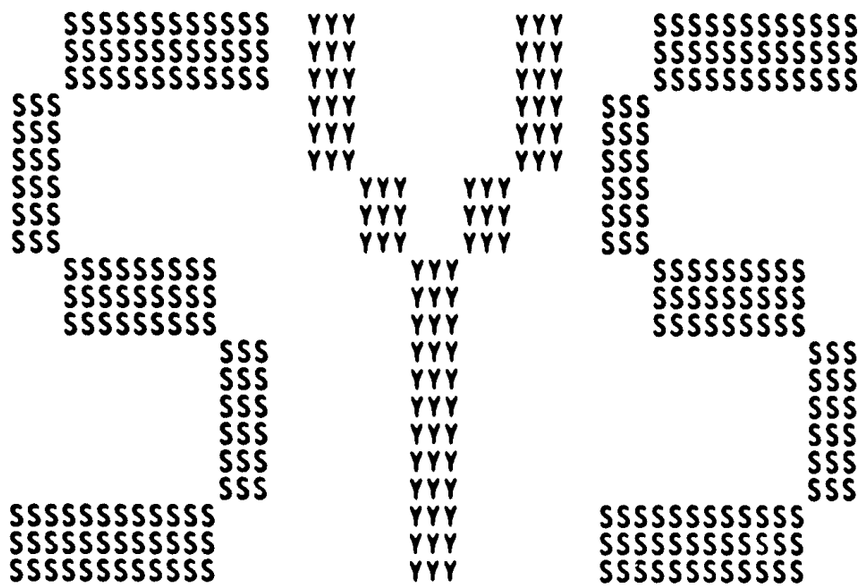

```
  SSSSSSSS  YY      YY    SSSSSSSS    CCCCCCCC  RRRRRRRR    EEEEEEEEEE  PPPPPPPP    RRRRRRRR      CCCCCCCC
  SSSSSSSS  YY      YY    SSSSSSSS    CCCCCCCC  RRRRRRRR    EEEEEEEEEE  PPPPPPPP    RRRRRRRR      CCCCCCCC
SS          YY      YY  SS          CC          RR      RR  EE          PP      PP  RR      RR  CC
SS          YY      YY  SS          CC          RR      RR  EE          PP      PP  RR      RR  CC
SS            YY  YY    SS          CC          RR      RR  EE          PP      PP  RR      RR  CC
SS            YY  YY    SS          CC          RR      RR  EE          PP      PP  RR      RR  CC
  SSSSSS        YY        SSSSSS    CC          RRRRRRRR    EEEEEEEE    PPPPPPPP    RRRRRRRR    CC
  SSSSSS        YY        SSSSSS    CC          RRRRRRRR    EEEEEEEE    PPPPPPPP    RRRRRRRR    CC
        SS      YY              SS  CC          RR  RR      EE          PP          RR  RR      CC
        SS      YY              SS  CC          RR  RR      EE          PP          RR  RR      CC
        SS      YY              SS  CC          RR    RR    EE          PP          RR    RR    CC                ....
        SS      YY              SS  CC          RR    RR    EE          PP          RR    RR    CC                ....
SSSSSSSS        YY      SSSSSSSS      CCCCCCCC  RR      RR  EEEEEEEEEE  PP          RR      RR    CCCCCCCC      ....
SSSSSSSS        YY      SSSSSSSS      CCCCCCCC  RR      RR  EEEEEEEEEE  PP          RR      RR    CCCCCCCC      ....


LL              IIIIII      SSSSSSSS
LL              IIIIII      SSSSSSSS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II        SSSSSS
LL                II        SSSSSS
LL                II              SS
LL                II              SS
LL                II              SS
LL                II              SS
LLLLLLLLLL      IIIIII    SSSSSSSS
LLLLLLLLLL      IIIIII    SSSSSSSS
```

Table of contents

```
0000     1          .TITLE  SYSCREPRC CREATE PROCESS SYSTEM SERVICE
0000     2          .IDENT  'V04-002'
0000     3
0000     4  ;*****************************************************************************
0000     5  ;*                                                                           *
0000     6  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     7  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     8  ;*  ALL RIGHTS RESERVED.                                                     *
0000     9  ;*                                                                           *
0000    10  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
0000    11  ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
0000    12  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
0000    13  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
0000    14  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
0000    15  ;*  TRANSFERRED.                                                             *
0000    16  ;*                                                                           *
0000    17  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
0000    18  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
0000    19  ;*  CORPORATION.                                                             *
0000    20  ;*                                                                           *
0000    21  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
0000    22  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    23  ;*                                                                           *
0000    24  ;*                                                                           *
0000    25  ;*****************************************************************************
0000    26
0000    27  ;++
0000    28  ; FACILITY: EXECUTIVE, SYSTEM SERVICES
0000    29  ;
0000    30  ; ABSTRACT: SYSCREPRC IMPLEMENTS THE CREATE PROCESS SYSTEM SERVICE.
0000    31  ;
0000    32  ; ENVIRONMENT: KERNEL MODE
0000    33  ;
0000    34  ; AUTHOR: R. HUSTVEDT    , CREATION DATE: 29-NOV-76
0000    35  ;
0000    36  ; MODIFIED BY:
0000    37  ;
0000    38  ;       V04-002 WMC0001         Wayne Cardoza           14-Sep-1984
0000    39  ;               Fix LJK0290 to only increment counts for detached processes.
0000    40  ;
0000    41  ;       V04-001 LJK0290         Lawrence J. Kenah       12-Sep-1984
0000    42  ;               Do not count process in SYS$GW_xJOBCNT cell until all error
0000    43  ;               conditions have been tested. This eliminates the need to
0000    44  ;               worry about the xJOBCNT cells along error paths.
0000    45  ;
0000    46  ;       V03-037 ACG0432         Andrew C. Goldstein,    9-Jul-1984  17:27
0000    47  ;               Add PRC$V_NOPASSWORD flag bit, remove DETACH priv
0000    48  ;               requirement from NOUAF, CLISPEC, and INTER flags.
0000    49  ;               Add and initialize JIB$L_ORG_BYTLM.
0000    50  ;
0000    51  ;       V03-036 PRD0100         Paul R. DeStefano       27-Jun-1984
0000    52  ;               Correct placement of label 10001$ in routine ACTIVATE.
0000    53  ;
0000    54  ;       V03-035 LJK0284         Lawrence J. Kenah       15-May-1984
0000    55  ;               Make SS$_INSSWAPSPACE into a useful error return instead
0000    56  ;               of a system crasher.
0000    57  ;
```

```
0000     58 ;        V03-034 MHB0155         Mark Bramhall           1-May-1984
0000     59 ;                Propagate PCB$(VIM)_SECAUDIT to new process's PCB.
0000     60 ;
0000     61 ;        V03-033 RAS0297         Ron Schaefer            18-Apr-1984
0000     62 ;                Put back bogus translation of SYS$DISK for compatibility
0000     63 ;                with past history.  Remove KPL0110 and KPL0111.
0000     64 ;
0000     65 ;        V03-032 KPL0111         Peter Lieberwirth       17-Apr-1984
0000     66 ;                The use of IOC$TRNADEVNAM, in V03-029, caused the device
0000     67 ;                string in PQB$T_DISK not to have a trailing colon.  This
0000     68 ;                caused PROCSTRT to create an invalid translation for
0000     69 ;                SYS$DISK.  Fix here by re-appending the colon.
0000     70 ;
0000     71 ;        V03-031 MHB0136         Mark Bramhall           12-Apr-1984
0000     72 ;                Add support for PRC$V_CLISPEC.
0000     73 ;
0000     74 ;        V03-030 MHB0134         Mark Bramhall           10-Apr-1984
0000     75 ;                Creators w/o JIBs => Username = SYSTEM, Account = binary nulls.
0000     76 ;                Account names with a leading binary null byte are special:
0000     77 ;                        <0><0>... => <0><0>...  All binary nulls stays as is.
0000     78 ;                        <0><x>... => <x>...< >  Others are shifted left one.
0000     79 ;                Move new spawn CLI information to PQB from P1 space.
0000     80 ;
0000     81 ;        V03-029 KPL0110         Peter Lieberwirth       31-Mar-1984
0000     82 ;                Translate SYS$DISK by calling IOC$TRANDEVNAM, which
0000     83 ;                uses $TRNLNM.  This call replaces obsolete $TRNLOG use.
0000     84 ;
0000     85 ;        V03-028 SRB0119         Steve Beckhardt         26-Mar-1984
0000     86 ;                Another round in the broken branch grame.
0000     87 ;
0000     88 ;        V03-027 LJK0263         Lawrence J. Kenah       29-Feb-1984
0000     89 ;                We're playing the broken branch game again. The various calls
0000     90 ;                to EXE$DEANONPAGED need more than word displacement.
0000     91 ;
0000     92 ;        V03-026 HH0002          Hai Huang               1-Feb-1984
0000     93 ;                Add job-wide mount support, i.e. initialize mount listhead
0000     94 ;                in JIB.
0000     95 ;
0000     96 ;        V03-025 LJK0259         Lawrence J. Kenah       23-Jan-1984
0000     97 ;                Fix incorrect register usage bug in PQB deallocation.
0000     98 ;
0000     99 ;        V03-024 LJK0258         Lawrence J. Kenah       18-Jan-1984
0000    100 ;                Fix bug introduced by LJK0257. Make JIB creation code handle
0000    101 ;                the case of the swapper, a process that does not own a JIB.
0000    102 ;
0000    103 ;        V03-023 ACG0385         Andrew C. Goldstein,    11-Jan-1984  18:39
0000    104 ;                Make MAXDETACH and MAXJOBS JIB fields words
0000    105 ;
0000    106 ;        V03-022 LJK0257         Lawrence J. Kenah       21-Dec-1983
0000    107 ;                Make changes to support larger PQB. Remove support for ACCOUNT
0000    108 ;                and USERNAME fields in P1 space. Use $TRNLNM to pick up
0000    109 ;                translation of SYS$DISK. Perform general cleanup.
0000    110 ;
0000    111 ;        V03-021 TMK0001         Todd M. Katz            12-Oct-1983
0000    112 ;                Add JTQUOTA (job-wide logical name table creation quota)
0000    113 ;                to the process quota block and as a quota in the quota list
0000    114 ;                for SYS$CREPRC. No special processing is required for this
```

```
0000   115 ;                   new quota item.
0000   116 ;
0000   117 ;       V03-020 JWT0138         Jim Teague              11-Oct-1983
0000   118 ;                   Fix broken branch to SCH$CHSE.
0000   119 ;
0000   120 ;       V03-019 RAS0181         Ron Schaefer            05-Sep-1983
0000   121 ;                   Convert creation of SYS$INPUT, SYS$OUTPUT, SYS$ERROR,
0000   122 ;                   and SYS$DISK logical names to use $CRELNM.
0000   123 ;
0000   124 ;       V03-018 WMC0008         Wayne Cardoza           01-Aug-1983
0000   125 ;                   New item list codes for logical name attributes.
0000   126 ;
0000   127 ;       V03-017 ACG0347         Andrew C. Goldstein,    1-Aug-1983  13:21
0000   128 ;                   Fix register use bug in ACG0335
0000   129 ;
0000   130 ;       V03-016 WMC0007         Wayne Cardoza           28-Jul-1983
0000   131 ;                   Move bumping of interactive and batch counts here.
0000   132 ;
0000   133 ;       V03-015 WMC0006         Wayne Cardoza           05-JUL-1983
0000   134 ;                   No privilege needed for interactive subprocess.
0000   135 ;
0000   136 ;       V03-014 CWH1010         CW Hobbs                20-Jun-1983
0000   137 ;                   Add a comment which calls attention to an order dependency
0000   138 ;                   in the call to EXE$IPID_TO_EPID.
0000   139 ;
0000   140 ;       V03-013 ACG0335         Andrew C. Goldstein,    9-May-1983  16:26
0000   141 ;                   Propagate default file protection in PCB; copy extended
0000   142 ;                   rights list to created process.
0000   143 ;
0000   144 ;       V03-012 WMC0005         Wayne Cardoza           27-Apr-1983
0000   145 ;                   Change PSECT of detach code.
0000   146 ;                   SS$_INSSWAPSPACE for no swap space.
0000   147 ;
0000   148 ;       V03-011 WMC0004         Wayne Cardoza           14-Apr-1983
0000   149 ;                   New STSFLG flags.
0000   150 ;
0000   151 ;       V03-010 WMC0003         Wayne Cardoza           31-Mar-1983
0000   152 ;                   Second half of the detach changes.
0000   153 ;
0000   154 ;       V03-009 WMC0002         Wayne Cardoza           10-Mar-1983
0000   155 ;                   Liberalized rules on creation of detached processes.
0000   156 ;
0000   157 ;       V03-008 ACG0318         Andrew C. Goldstein,    8-Mar-1983  20:27
0000   158 ;                   Initialize new ARB fields in created process
0000   159 ;
0000   160 ;       V03-007 MTR0001         Michael T. Rhodes       28-Feb-1983
0000   161 ;                   Change the privilege requirements for setting the initial
0000   162 ;                   process state flags to require only the creator to have
0000   163 ;                   privilege.
0000   164 ;
0000   165 ;       V03-006 CWH1002         CW Hobbs                24-Feb-1983
0000   166 ;                   Create new extended process ident PCB$L_EPID.  Return the
0000   167 ;                   extended pid to the pidadr argument if specifed.  The
0000   168 ;                   extended pid of a subprocess owner propagates to the
0000   169 ;                   new PCB$L_EOWNER field.
0000   170 ;
0000   171 ;       V03-005 CWH1001         CW Hobbs                15-Feb-1983
```

```
0000   172 ;                   Change from sequential to round-robin PIX allocation.
0000   173 ;
0000   174 ;       V03-004 WMC0001         Wayne Cardoza           18-Oct-1982
0000   175 ;                   Add support for item list argument for page file control.
0000   176 ;
0000   177 ;       V03-003 LJK48272        Lawrence J. Kenah       10-Aug-1982
0000   178 ;                   Insure that PCB$L_JIB is clear before allocating PQB
0000   179 ;                   in case PQB allocation fails. Error code assumes that
0000   180 ;                   contents of JIB field are always valid if nonzero.
0000   181 ;                   Remove $PRDEF and $SSDEF calls.
0000   182 ;
0000   183 ;       V03-002 LJK0169         Lawrence J. Kenah       2-Jun-1982
0000   184 ;                   Insure that revised CPU limit is stored in PQB along error
0000   185 ;                   paths. Use ROTL to perform unsigned divide by two.
0000   186 ;
0000   187 ;--
```

```
                     0000   189          .SBTTL  DECLARATIONS
                     0000   190 ;
                     0000   191 ; INCLUDE FILES:
                     0000   192 ;
                     0000   193          $ACLDEF                         ; DEFINE ACL BLOCK
                     0000   194          $ARBDEF                         ; DEFINE ACCESS RIGHTS BLOCK
                     0000   195          $DYNDEF                         ; DATA STRUCTURE IDENTIFIERS
                     0000   196          $IPLDEF                         ; DEFINE INTERRUPT PRIORITY LEVELS
                     0000   197          $JIBDEF                         ; DEFINE JOB INFORMATION BLOCK
                     0000   198          $LNMDEF                         ; DEFINE LNM OFFSETS
                     0000   199          $LNMSTRDEF                      ; DEFINE LNM BLOCK OFFSETS
                     0000   200          $PCBDEF                         ; DEFINE PCB OFFSETS
                     0000   201          $PFNDEF                         ; DEFINE PFN CONSTANTS
                     0000   202          $PHDDEF                         ; DEFINE PHD OFFESTS
                     0000   203          $PQBDEF                         ; DEFINE PROCESS QUOTA BLOCK
                     0000   204          $PQLDEF                         ; DEFINE PROCESS QUOTA LIST CODES
                     0000   205          $PRCDEF                         ; DEFINE $CREPRC STATUS FLAGS
                     0000   206          $PRIDEF                         ; DEFINE PRIORITY INCREMENT CLASSES
                     0000   207          $PRVDEF                         ; DEFINE PRIVILEGE BITS
                     0000   208
                     0000   209 ;***** Temporary definitions for ARB cells until SDL is fixed to
                     0000   210 ;***** expand $ARBDEF correctly.
                     0000   211 ;
00000030             0000   212 ARB$R_RIGHTSDESC=48
00000038             0000   213 ARB$R_LOCALRIGHTS=56
                     0000   214 ;
                     0000   215 ; MACROS:
                     0000   216 ;
                     0000   217
                     0000   218 ;
                     0000   219 ; MACRO TO CREATE STSFLG MAPPING AND PRIVILEGE CHECK TABLES:
                     0000   220 ;       STSNAM          OPTIONAL STATUS BIT NAME TO SET IN PCB
                     0000   221 ;       PRVNAM          OPTIONAL REQUIRED PRIVILEGE BIT NAME
                     0000   222 ;       NOSUBPRV        OPTIONAL NO PRIVILEGE REQUIRED IF SUBPROCESS FLAG
                     0000   223 ;
                     0000   224          .MACRO  STSFLAG STSNAM,PRVNAM,NOSUBPRV
                     0000   225          .IF     B,PRVNAM
                     0000   226          .BYTE   -1
                     0000   227          .IFF
                     0000   228          .IF     B,NOSUBPRV
                     0000   229          .BYTE   PRV$V_'PRVNAM
                     0000   230          .IFF
                     0000   231          .BYTE   PRV$V_'PRVNAM ! ^X80
                     0000   232          .ENDC
                     0000   233          .ENDC
                     0000   234          .IF     B,STSNAM
                     0000   235          .BYTE   -1
                     0000   236          .IFF
                     0000   237          .BYTE   PCB$V_'STSNAM
                     0000   238          .ENDC
                     0000   239          .ENDM   STSFLAG
                     0000   240
                     0000   241 ;
                     0000   242 ; MACRO TO CALL STRING MOVING AND VERIFICATION ROUTINE
                     0000   243 ;
                     0000   244          .MACRO  MOVSTRING LIM=15,SRC,DST
                     0000   245          BSBW    MOVSTR                  ; CALL MOVE SUBROUTINE
```

```
                     0000   246          .BYTE   LIM                             ; COUNT LIMIT
                     0000   247          .BYTE   SRC@-2                          ; SOURCE OFFSET FROM AP
                     0000   248          .WORD   PQB$T_DST                       ; DESTINATION OFFSET IN PQB
                     0000   249          .ENDM   MOVSTRING
                     0000   250
                     0000   251 ;
                     0000   252 ; EQUATED SYMBOLS:
                     0000   253 ;
                     0000   254
00000004             0000   255 PIDADR=4                                        ; PID ADDRESS
00000008             0000   256 IMAGE=8                                         ; IMAGE NAME
0000000C             0000   257 INPUT=12                                        ; INPUT LOGICAL NAME DESCRIPTOR
00000010             0000   258 OUTPUT=16                                       ; OUTPUT LOGICAL NAME DESCRIPTOR
00000014             0000   259 ERROR=20                                        ; ERROR LOGICAL NAME DESCRIPTOR
00000018             0000   260 PRVADR=24                                       ; PRIVILEGE MASK ADDRESS
0000001C             0000   261 QUOTA=28                                        ; QUOTA BUFFER POINTER
00000020             0000   262 PRCNAM=32                                       ; PROCESS NAME DESCRIPTOR
00000024             0000   263 BASPRI=36                                       ; BASE PRIORITY
00000028             0000   264 UIC=40                                          ; UIC
0000002C             0000   265 MBXUNT=44                                       ; MAILBOX UNIT NUMBER
00000030             0000   266 STSFLG=48                                       ; STATUS FLAG MASK
00000034             0000   267 ITMLST=52                                       ; ITEM LIST
                     0000   268
00000000             0000   269 PQL_V_DEDUCT=0                                 ; DEDUCTIBLE QUOTA FLAG
0000000D             0000   270 ITMLST_ARG=13                                   ; ARGUMENT NUMBER FOR THE ITEM LIST
                     0000   271
FFFFFFFC             0000   272 CURPCB = -4                                     ; OFFSET FROM FP TO SAVED R4
                     0000   273
                     0000   274 ;
                     0000   275 ; OWN STORAGE:
                     0000   276 ;
                     0000   277
00000000             0000   278          .PSECT  Y$EXEPAGED,BYTE                ; PAGEABLE PSECT
                     0000   279
                     0000   280          ASSUME  PRC$V_SSRWAIT  EQ  0
                     0000   281          ASSUME  PRC$V_SSFEXCU  EQ  1
                     0000   282          ASSUME  PRC$V_PSWAPM   EQ  2
                     0000   283          ASSUME  PRC$V_NOACNT   EQ  3
                     0000   284          ASSUME  PRC$V_BATCH    EQ  4
                     0000   285          ASSUME  PRC$V_HIBER    EQ  5
                     0000   286          ASSUME  PRC$V_NOUAF    EQ  6
                     0000   287          ASSUME  PRC$V_NETWRK   EQ  7
                     0000   288          ASSUME  PRC$V_DISAWS   EQ  8
                     0000   289          ASSUME  PRC$V_DETACH   EQ  9
                     0000   290          ASSUME  PRC$V_INTER    EQ  10
                     0000   291          ASSUME  PRC$V_IMGDMP   EQ  11
                     0000   292          ASSUME  PRC$V_CLISPEC  EQ  12
                     0000   293          ASSUME  PRC$V_NOPASSWORD EQ  13
                     0000   294
                     0000   295 STSFLGTBL:                                     ; TRANSLATION TABLE FOR STATUS FLAG BITS
                     0000   296          STSFLAG SSRWAIT                        ; BIT 0 => RESOURCE WAIT
                     0002   297          STSFLAG SSFEXCU                        ; BIT 1 => SYSTEM SERVICE FAIL EXCEPTION
                     0004   298                                                 ;          FOR USER MODE
                     0004   299          STSFLAG PSWAPM,PSWAPM                  ; BIT 2 => PROCESS SWAP MODE
                     0006   300          STSFLAG NOACNT,NOACNT                  ; BIT 3 => NO ACCOUNTING MESSAGE
                     0008   301          STSFLAG BATCH,DETACH                   ; BIT 4 => BATCH
                     000A   302          STSFLAG HIBER                          ; BIT 5 => HIBERNATE BEFORE CALLING
```

```
                          000C   303                                     ;           INITIAL IMAGE IN PROCSTRT
                          000C   304          STSFLAG LOGIN              ; BIT 6 => LOGIN WITHOUT READING AUTH FILE
                          000E   305          STSFLAG NETWRK,DETACH      ; BIT 7 => NETWORK
                          0010   306          STSFLAG DISAWS             ; BIT 8 => DISABLE WORKING SET ADJUST
                          0012   307          STSFLAG                    ; BIT 9 => DETACH
                          0014   308          STSFLAG INTER              ; BIT 10 => INTERACTIVE
                          0016   309          STSFLAG                    ; BIT 11 => IMAGE DUMP
                          0018   310          STSFLAG                    ; BIT 12 => PASS ON CLI SPECIFICATIONS
                          001A   311          STSFLAG                    ; BIT 13 => NO USERNAME DIALOGUE
                 0000000E 001C   312 STSFLGCNT=<.-STSFLGTBL>@-1          ; NUMBER OF STATUS FLAGS
                          001C   313
                          001C   314 ; THE FOLLOWING TEXT FIELDS ARE USED WHEN THE CREATING PROCESS (SUCH AS THE
                          001C   315 ; SWAPPER) DOES NOT HAVE A JIB.
                          001C   316
                          001C   317 DEFAULT_NAMES:
        4D 45 54 53 59 53 001C   318          .ASCII  'SYSTEM'           ; Username is SYSTEM, blank padded
        20'20'20'20'20'20' 0022   319          .BYTE   ^A' ' [JIB$S_USERNAME - <. - DEFAULT_NAMES>]
  00'00'00'00'00'00'00'00' 0028   320          .BYTE   0 [JIB$S_ACCOUNT]  ; Account name is binary nulls
                          0030   321          ASSUME  <. - DEFAULT_NAMES> EQ <JIB$S_USERNAME + JIB$S_ACCOUNT>
                          0030   322
                          0030   323 ; LOGICAL NAME DATA FOR USE IN TRANSLATING SYS$DISK
                          0030   324
49 46 24 4D 4E 4C 00000038'0030   325 LNM_TBL:.ASCID  \LNM$FILE_DEV\
        56 45 44 5F 45 4C 003E
                 00000103 0044   326 LNM_ATTR = ^X0103
                          0044   327
```

```
0044    329          .SBTTL  EXE$CREPRC - CREATE PROCESS SYSTEM SERVICE
0044    330 ;++
0044    331 ; FUNCTIONAL DESCRIPTION:
0044    332 ;       EXE$CREPRC CREATES A NEW PROCESS ACCORDING TO THE
0044    333 ;       SUPPLIED PARAMETERS.  THE NEW PROCESS MAY BE EITHER A SUB-PROCESS
0044    334 ;       OR AN INDEPENDENT, DETACHED PROCESS.
0044    335 ;
0044    336 ; CALLING SEQUENCE:
0044    337 ;       CALLG   ARGLIST,EXE$CREPRC
0044    338 ;
0044    339 ; INPUT PARAMETERS:
0044    340 ;       PIDADR(AP) - ADDRESS AT WHICH TO RETURN PID OF CREATED PROCESS
0044    341 ;       IMAGE(AP)  - ADDRESS OF IMAGE NAME STRING DESCRIPTOR
0044    342 ;       INPUT(AP)  - ADDRESS OF INPUT NAME STRING DESCRIPTOR
0044    343 ;       OUTPUT(AP) - ADDRESS OF OUTPUT NAME STRING DESCRIPTOR
0044    344 ;       ERROR(AP)  - ADDRESS OF ERROR LOGICAL NAME STRING DESCRIPTOR
0044    345 ;       PRVADR(AP) - ADDRESS OF PRIVILEGE MASK FOR CREATED PROCESS
0044    346 ;       QUOTA(AP)  - POINTER TO QUOTA BUFFER
0044    347 ;       PRCNAM(AP) - ADDRESS OF PROCESS NAME STRING DESCRIPTOR
0044    348 ;       BASPRI(AP) - BASE PRIORITY FOR CREATED PROCESS
0044    349 ;       UIC(AP)    - UIC FOR CREATED PROCESS(0 => SUB-PROCESS)
0044    350 ;       MBXUNT(AP) - MAILBOX UNIT NUMBER FOR TERMINATION MESSAGES
0044    351 ;       STSFLG(AP) - STATUS FLAG SETTINGS FOR CREATED PROCESS
0044    352 ;       ITMLST(AP) - ITEM LIST
0044    353 ;       R4 - ADDRESS OF CURRENT PROCESS CONTROL BLOCK
0044    354 ;
0044    355 ;                          BIT     MEANING
0044    356 ;                          ---     -------
0044    357 ;                           0      RESOURCE WAIT DISABLE
0044    358 ;                           1      SYSTEM SERVICE FAIL EXCEPTION ENABLE
0044    359 ;                           2      PROCESS SWAP MODE
0044    360 ;                           3      ACCOUNTING MESSAGE DISABLE
0044    361 ;                           4      BATCH INDICATOR
0044    362 ;                           5      HIBERNATE BEFORE CALLING INITIAL IMAGE
0044    363 ;                           6      BYPASS LOGIN VERIFICATION FOR DETACHED
0044    364 ;                                  PROCESS.
0044    365 ;                           7      NETWORK INDICATOR
0044    366 ;                           8      DISABLE WORKING SET ADJUSTMENT
0044    367 ;                           9      DETACHED PROCESS
0044    368 ;                          10      INTERACTIVE INDICATOR
0044    369 ;                          11      IMAGE DUMP ON FATAL ABORT
0044    370 ;                          12      PASS ON CLI SPECIFICATIONS
0044    371 ;
0044    372 ;
0044    373 ; OUTPUT PARAMETERS:
0044    374 ;       R0  -  COMPLETION STATUS CODE
0044    375 ;       @PIDADR(AP) - PROCESS ID (PID) OF CREATED PROCESS
0044    376 ;
0044    377 ; COMPLETION CODES:
0044    378 ;       SS$_NORMAL    -   SUCCESSFUL COMPLETION
0044    379 ;       SS$_ACCVIO    -   ACCESS VIOLATION
0044    380 ;       SS$_DUPLNAM   -   DUPLICATE PROCESS NAME
0044    381 ;       SS$_EXQUOTA   -   EXCEEDED QUOTA
0044    382 ;       SS$_INSFMEM   -   INSUFFICIENT MEMORY AVAILABLE
0044    383 ;       SS$_IVLOGNAM  -   INVALID LOGICAL NAME
0044    384 ;       SS$_IVQUOTAL  -   INVALID QUOTA LIST
0044    385 ;       SS$_IVSTSFLG  -   INVALID STATUS FLAG ARGUMENT
```

```
                     0044   386 ;       SS$_NOPRIV   -   NO PRIVILEGE FOR SPECIFIED OPERATION
                     0044   387 ;
                     0044   388 ; SIDE EFFECTS:
                     0044   389 ;       IF NO ERRORS ARE DETECTED, A NEW PROCESS WILL HAVE BEEN ACTIVATED
                     0044   390 ;       AND MARKED NON-RESIDENT.  THE INITIAL INSWAP FOR THIS PROCESS
                     0044   391 ;       WILL BE FROM THE SHELL PROCESS.  EXECUTION FOR THIS PROCESS
                     0044   392 ;       BEGINS IN THE ROUTINE EXE$PROCSTRT WHICH WILL MOVE THE INFORMATION
                     0044   393 ;       FROM THE PROCESS QUOTA BLOCK TO THE APPROPRIATE LOCATIONS
                     0044   394 ;       IN THE PROCESS CONTEXT.  CONTROL WILL THEN BE GIVEN TO THE
                     0044   395 ;       SPECIFIED IMAGE.
                     0044   396 ;
                     0044   397 ;--
                     0044   398
                     0044   399             .ENABL  LSB
                     0044   400 EXE$CREPRC::                          ; CREATE PROCESS SYSTEM SERVICE
                OFFC 0044   401             .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>     ; SAVE ALL REGISTERS
                  54 DD     0046   402             PUSHL   R4                          ; SAVE CREATOR'S PCB ADDRESS
               04 AC DD     0048   403             PUSHL   PIDADR(AP)                  ; SAVE PID RETURN ADDRESS
                  07 13     004B   404             BEQL    10$                         ; NONE, NO PROBE
                     004D   405             IFNOWRT #4,@(SP),20$                ; CHECK FOR WRITABLE
             5B 28 AC DO     0054   406 10$:         MOVL    UIC(AP),R11                 ; FETCH UIC FOR CREATED PROCESS
                  2B 13     0058   407             BEQL    29$                         ; NOT SPECIFIED
           00BC C4 5B D1     005A   408             CMPL    R11,PCB$L_UIC(R4)           ; IS IT SAME UIC AS CREATOR
                  2E 13     005F   409             BEQL    30$                         ; NO PRIVILEGE NEEDED
                     0061   410             IFPRIV  DETACH,30$                  ; FULL DETACH PRIVILEGE
                     0067   411             IFPRIV  CMKRNL,30$                  ; OR CHANGE MODE TO KERNEL
           50 0000'8F 3C     006D   412             MOVZWL  #SS$_NOPRIV,R0              ; NO, SET ERROR CODE
                  04 0072   413 15$:        RET                                 ; RETURN
                     0073   414
           50 0000'8F 3C     0073   415 20$:         MOVZWL  #SS$_ACCVIO,R0              ; SET ERROR CODE FOR ACCESS VIOLATION
                  04 0078   416             RET                                 ; RETURN
                     0079   417
                     0079   418 ; FAST ALLOCATION OF PQB FAILED. DO IT THE HARD WAY
                     0079   419
                085E 30     0079   420 26$:         BSBW    ALLOCPQB                    ; ALLOCATE ANOTHER PQB
              59 52 DO      007C   421             MOVL    R2,R9                       ; ASSUME THAT IT SUCCEEDED
                39 50 E8    007F   422             BLBS    R0,50$                      ; RESUME CREATION IF SUCCESSFUL
                07B9 31     0082   423             BRW     ABRT2                       ; ERROR PATH FOR PQB ALLOCATION FAILURE
                     0085   424
          05 30 AC 09 E1    0085   425 29$:         BBC     #PRC$V_DETACH,STSFLG(AP),30$
           5B 00BC C4 D0    008A   426             MOVL    PCB$L_UIC(R4),R11           ; DETACHED PROCESS (SAME UIC) REQUESTED
            00000000'EF 16  008F   427 30$:         JSB     EXE$ALLOCPCB                ; ALLOCATE PCB, WAIT IF NECESSARY
                  DA 50 E9  0095   428             BLBC    R0,15$                      ; RETURN ERROR IF FAILURE
               5A 52 DO     0098   429             MOVL    R2,R10                      ; SAVE ADDRESS OF NEW PCB
OC AA 0114 8F 00 6E 00      009B   430             MOVC5   #0,(SP),#0,#<PCB$C_LENGTH-12>,12(R10)   ; CLEAR PCB
              54 FC AD DO   00A4   431             MOVL    CURPCB(FP),R4               ; RESTORE CREATOR PCB ADDRESS
           F7FFFFFF 8F CB   00A8   432             BICL3   #^C<PCB$M_SECAUDIT>,-       ; PROPAGATE (ONLY) MANDATORY AUDITING
            24 AA 24 A4     00AE   433                     PCB$L_STS(R4),PCB$L_STS(R10) ;  FROM PARENT TO NEW PCB
        59 00000000'FF 0F   00B2   434             REMQUE  @EXE$GL_PQBFL,R9            ; ATTEMPT FAST PQB ALLOCATION
                  BE 1D     00B9   435             BVS     26$                         ; OUT IF LINE IF FAILURE
                     00BB   436 ;
                     00BB   437 ;       INITIALIZE NEW PCB
                     00BB   438 ;
                  5B D5     00BB   439 50$:         TSTL    R11                         ; IS THIS A DETACHED CREATE?
                  26 12     00BD   440             BNEQ    51$                         ; YES, GO ALLOCATE A NEW JIB
           52 0080 C4 DO    00BF   441             MOVL    PCB$L_JIB(R4),R2            ; GET JIB ADDRESS OF PARENT
           0080 CA 52 DO    00C4   442             MOVL    R2,PCB$L_JIB(R10)           ; SAVE POINTER TO JIB IN OFFSPRING
```

```
                44 A2   B6  00C9   443          INCW    JIB$W_PRCCNT(R2)          ; ADD ANOTHER SUBPROCESS
        3C A2  00000000'8F  C2  00CC   444          SUBL    #SWP$C_SHELLPFIL,JIB$L_PGFLCNT(R2); CHARGE FOR SHELL PAGES
                     07     19  00D4   445          BLSS    23$                       ; BR IF OUT OF PAGE FILE QUOTA
           46 A2    44 A2   B1  00D6   446          CMPW    JIB$W_PRCCNT(R2),JIB$W_PRCLIM(R2) ; OVER LIMIT?
                     4A     1B  00DB   447          BLEQU   56$                       ; IF LEQU NO, CONTINUE
            50  0000'8F     3C  00DD   448 23$:     MOVZWL  #SS$_EXQUOTA,R0           ; SET ERROR STATUS
                   075B     31  00E2   449 24$:     BRW     ABORT                     ; YES ABORT
                            00E5   450
             00000000'EF    16  00E5   451 51$:     JSB     EXE$ALLOCJIB              ; ALLOCATE JIB FOR MASTER PROCESS
                   F4 50    E9  00EB   452          BLBC    R0,24$                    ; BR IF NO SPACE AVAILABLE
                62   62     DE  00EE   453          MOVAL   JIB$L_MTLFL(R2),JIB$L_MTLFL(R2) ; INITIALIZE MOUNT LISTHEAD
             04 A2   62     DE  00F1   454          MOVAL   JIB$L_MTLFL(R2),JIB$L_MTLBL(R2) ;
           0080 CA   52     D0  00F5   455          MOVL    R2,PCB$L_JIB(R10)         ; SAVE POINTER TO JIB
                            00FA   456
                            00FA   457 ; THE USERNAME AND ACCOUNT FIELDS OF THE CREATOR'S JIB ARE MOVED INTO THE
                            00FA   458 ; NEW JIB. THE REST OF THE JIB IS CLEARED. THE TWO ASSUME STATEMENTS INSURE
                            00FA   459 ; THAT THE JIB LAYOUT ALLOWS ALL OF THESE SHENANIGANS TO BE ACCOMPLISHED
                            00FA   460 ; WITH A SINGLE MOVC5 INSTRUCTION.
                            00FA   461
                            00FA   462          ASSUME JIB$T_USERNAME EQ 12
                            00FA   463          ASSUME JIB$T_ACCOUNT EQ <12 + JIB$S_USERNAME>
                            00FA   464
            53   0080 C4    D0  00FA   465          MOVL    PCB$L_JIB(R4),R3          ; GET JIB ADDRESS OF CREATOR
                      05    12  00FF   466          BNEQ    53$                       ; JIB EXISTS, GO USE IT
                            0101   467
                            0101   468 ; THE SWAPPER PROCESS DOES NOT HAVE A JIB. LOAD R3 WITH THE ADDRESS OF SOME
                            0101   469 ; TEXT FIELDS THAT LOAD THE NEW JIB WITH A DEFAULT USERNAME AND ACCOUNT.
                            0101   470
            53   FF0B CF    9E  0101   471          MOVAB   DEFAULT_NAMES-JIB$T_USERNAME,R3 ; GET STRING ADDRESS
                      14    2C  0106   472 53$:     MOVC5   #<JIB$S_USERNAME + JIB$S_ACCOUNT>,-
                   0C A3      0108   473                  JIB$T_USERNAME(R3),-      ; CREATOR'S JIB IS SOURCE
                      00      010A   474                  #0,-                      ; FILL THE REST WITH ZEROS
                   0068 8F    010B   475                  #<JIB$C_LENGTH-12>,-
                   0C A2      010E   476                  JIB$T_USERNAME(R2)        ; NEW JIB IS DESTINATION
                            0110   477
                            0110   478 ; Check the account name for a leading binary null.  Account names with
                            0110   479 ; a leading binary null are special, reserved to DIGITAL, account names.
                            0110   480 ; The following account name combinations are possible:
                            0110   481 ;       <x>...          Normal account name; left as is.
                            0110   482 ;       <0><0>...       Initial startup account name; left as is.
                            0110   483 ;       <0><x>...       Special account name; shifted left one place.
                            0110   484
            52   0080 CA    D0  0110   485          MOVL    PCB$L_JIB(R10),R2         ; Re-address offspring's JIB
                   18 A2    95  0115   486          TSTB    JIB$T_ACCOUNT(R2)         ; A leading binary null byte?
                      0D    12  0118   487          BNEQ    56$                       ; Nope, leave account name as is
                   19 A2    95  011A   488          TSTB    JIB$T_ACCOUNT+1(R2)       ; Anything after the null to shift?
                      08    13  011D   489          BEQL    56$                       ; Nope, leave account name alone
                      07    2C  011F   490          MOVC5   #JIB$S_ACCOUNT-1,-        ; Move all but the first byte
                   19 A2      0121   491                  JIB$T_ACCOUNT+1(R2),-     ;  of the account name,
                      20      0123   492                  #^A' ',-                  ;   padding with a blank,
                      08      0124   493                  #JIB$S_ACCOUNT,-          ;    into the full
                   18 A2      0125   494                  JIB$T_ACCOUNT(R2)         ;     account name
                            0127   495
         10 AA     10 AA    DE  0127   496 56$:     MOVAL   PCB$L_ASTQFL(R10),PCB$L_ASTQFL(R10)     ; SET UP AST Q HEADER
         14 AA     10 AA    DE  012C   497          MOVAL   PCB$L_ASTQFL(R10),PCB$L_ASTQBL(R10)     ;
            0D AA     0F    90  0131   498          MOVB    #^X0F,PCB$B_ASTEN(R10)                 ; SET ALL AST ENABLES
      0104 CA   0104 CA    DE  0135   499          MOVAL   PCB$L_LOCKQFL(R10),PCB$L_LOCKQFL(R10)  ; SET UP LOCK Q HEADER
```

```
                0108 CA   0104 CA   DE  013C   500          MOVAL   PCB$L_LOCKQFL(R10),PCB$L_LOCKQBL(R10)
                     54     FC AD   D0  0143   501          MOVL    CURPCB(FP),R4                       ; RESTORE CREATOR PCB ADDRES
                   56   00AC C4   D0  0147   502          MOVL    PCB$Q_PRIV+ARB$L_RIGHTSLIST+8(R4),R6    ; GET EXTENDED RIGHTS LIST,
                0114 CA   0114 C4   D0  014C   503          MOVL    PCB$L_DEFPROT(R4),PCB$L_DEFPROT(R10)    ; COPY DEFAULT FILE PROTECTI
                     09 5E    1F   E1  0153   504          BBC     #31,SP,57$                          ; BRANCH IF NOT INITIAL CREA
          0114 CA    00000000'EF   3C  0157   505          MOVZWL  SYS$GW_FILEPROT,PCB$L_DEFPROT(R10)  ; SET SYSTEM FILE PROTECTION
    0084 CA   0C84 C4   0078 8F   28  0160   506 57$:      MOVC3   #ARB$C_LENGTH,PCB$Q_PRIV(R4),PCB$Q_PRIV(R10) ; COPY ENTIRE ARB
                0088 CA   0084 CA   9E  016A   507          MOVAB   PCB$Q_PRIV(R10),PCB$L_ARB(R10)      ; SET ADDRESS OF ARB
                          00B4 CA   9E  0171   508          MOVAB   PCB$Q_PRIV+ARB$R_RIGHTSDESC(R10),-
                          00A4 CA         0175   509                  PCB$Q_PRIV+ARB$L_RIGHTSLIST(R10)    ; ADDR OF LOCAL RIGHTS DESC
                          00BC CA   9E  0178   510          MOVAB   PCB$Q_PRIV+ARB$R_LOCALRIGHTS(R10),-
                          00B8 CA         017C   511                  PCB$Q_PRIV+ARB$R_RIGHTSDESC+4(R10)  ; ADDR OF LOCAL RIGHTS LIST
                               56   D5  017F   512          TSTL    R6                                  ; SEE IF EXTENDED RIGHTS LIS
                               25   13  0181   513          BEQL    59$                                 ; BRANCH IF NONE
                          00AC CA   D4  0183   514          CLRL    PCB$Q_PRIV+ARB$L_RIGHTSLIST+8(R10)  ; MARK NONE ALLOCATED YET
                       51   08 A6   3C  0187   515          MOVZWL  ACL$W_SIZE(R6),R1                   ; GET SIZE
                    00000000'EF    16  018B   516          JSB     EXE$ALLOCBUF                        ; AND ALLOCATE A NEW ONE
                          03 50    E8  0191   517          BLBS    R0,58$                              ; KEEP GOING IF SUCCESS
                             06A9  31  0194   518          BRW     ABORT                               ; ABORT CREATION
                                   0197   519
                          57   52   D0  0197   520 58$:      MOVL    R2,R7                               ; SAVE ADDRESS
                    00AC CA    52   D0  019A   521          MOVL    R2,PCB$Q_PRIV+ARB$L_RIGHTSLIST+8(R10) ; STORE IN PCB
                    67   66    51   28  019F   522          MOVC3   R1,(R6),(R7)                        ; COPY RIGHTS LIST CONTENTS
                 04 A7    0C A7   9E  01A3   523          MOVAB   12(R7),4(R7)                        ; SET DESCRIPTOR POINTER
                 32 AA    2C AC   B0  01A8   524 59$:      MOVW    MBXUNT(AP),PCB$W_TMBU(R10)          ; TERMINATION MAILBOX UNIT
           36 AA    00000000'EF   B0  01AD   525          MOVW    SWP$GL_SHELLSIZ,PCB$W_PPGCNT(R10)   ; AND PROCESS PAGE COUNT
                                   01B5   526 ;
                                   01B5   527 ;          INITIALIZE QUOTA BUFFER
                                   01B5   528 ;
0C A9    08BC 8F    00    6E    00  2C   01B5   529          MOVC5   #0,(SP),#0,#<PQB$C_LENGTH-12>,12(R9)    ; CLEAR QUOTA BUFFER
                                   01BE   530          MOVSTRING       LIM=15,-            ; MOVE PROCESS NAME TO PQB TO VALIDATE
                                   01BE   531                  SRC=PRCNAM,-                ; FROM PROCESS NAME DESCRIPTOR
                                   01BE   532                  DST=IMAGE                   ; TO IMAGE SLOT IN PQB
                   56   07C8 C9   9A  01C5   533          MOVZBL  PQB$T_IMAGE(R9),R6          ; FETCH STRING LENGTH FROM BUFFER
                             0D    13  01CA   534          BEQL    60$                         ; NO PROCESS NAME, NULL STRING
                             56    D6  01CC   535          INCL    R6                          ; INCLUDE COUNT BYTE
           70 AA    07C8 C9    56   28  01CE   536          MOVC3   R6,PQB$T_IMAGE(R9),PCB$T_LNAME(R10) ; SET INTO NEW PCB
                          07C8 C9    7C  01D5   537          CLRQ    PQB$T_IMAGE(R9)             ; CLEAR IMAGE NAME LENGTH
                     54     FC AD   D0  01D9   538 60$:      MOVL    CURPCB(FP),R4               ; RESTORE PCB ADDRESS
                     50     6C B4   7D  01DD   539          MOVQ    @PCB$L_PHD(R4),R0           ; GET PRIVILEGE MASK
                          52   50    7D  01E1   540          MOVQ    R0,R2                       ; ASSUME SAME PRIVILEGES
                     57     18 AC   D0  01E4   541          MOVL    PRVADR(AP),R7               ; FETCH ADDRESS OF PRIVILEGE MASK
                                1B   13  01E8   542          BEQL    70$                         ; NONE SPECIFIED
                                     01EA   543          IFNORD  #8,(R7),125$                ; ERROR IF NOT READABLE
                          52   67    7D  01F0   544          MOVQ    (R7),R2                     ; FETCH NEW PRIVILEGE MASK
                                     01F3   545          IFPRIV  SETPRV,70$                  ; SKIP MINIMIZATION IF SETPRV
                          50   50    D2  01F9   546          MCOML   R0,R0                       ; INVERT FOR BIT CLR
                          51   51    D2  01FC   547          MCOML   R1,R1                       ;
                          52   50    CA  01FF   548          BICL    R0,R2                       ; MINIMIZE PRIVILEGES
                          53   51    CA  0202   549          BICL    R1,R3                       ; IF CALLER IS NOT PRIVILEGED ENOUGH
                          69   52    7D  0205   550 70$:      MOVQ    R2,PQB$Q_PRVMSK(R9)         ; SET PRIVILEGES IN BUFFER
           50    24 AC    05    00    EF  0208   551          EXTZV   #0,#5,BASPRI(AP),R0         ; FETCH DESIRED BASE PRIORITY
                 50    1F    50    C3  020E   552          SUBL3   R0,#31,R0                   ; CONVERT TO INTERNAL PRIORITY
                                     0212   553          IFPRIV  SETPRI,100$                 ; SKIP MINIMIZATION IF PRIVILEGE
                    2F A4    50    91  0218   554          CMPB    R0,PCB$B_PRIB(R4)           ; OTHERWISE MINIMIZE WITH CALLERS BASE
                                04   1E  021C   555          BGEQU   100$                        ; GOOD VALUE, USE IT
                     50    2F A4    90  021E   556          MOVB    PCB$B_PRIB(R4),R0           ; NO, USE CALLERS BASE PRIORITY
```

```
                2F AA    50    90  0222    557 100$:   MOVB    R0,PCB$B_PRIB(R10)      ; SET IN NEW PCB
                0B AA    50    90  0226    558          MOVB    R0,PCB$B_PRI(R10)       ; AS BOTH CURRENT AND BASE PRIORITY
             00BC CA     5B    D0  022A    559          MOVL    R11,PCB$L_UIC(R10)      ; STORE UIC FOR NEW PROCESS
                         11    12  022F    560          BNEQ    110$                    ; BR IF IT WAS SPECIFIED
      00BC CA    00BC C4    D0  0231    561          MOVL    PCB$L_UIC(R4),PCB$L_UIC(R10)    ; OTHERWISE USE UIC OF CREATOR
          1C AA     60 A4    D0  0238    562          MOVL    PCB$L_PID(R4),PCB$L_OWNER(R10)  ; AND INDICATE SUBPROCESS
          68 AA     64 A4    D0  023D    563          MOVL    PCB$L_EPID(R4),PCB$L_EOWNER(R10); AND ALSO COPY THE EXTENDED PID
       57    00000000'EF    3C  0242    564 110$:    MOVZWL  SCH$GL_MAXPIX,R7        ; INITIALIZE INDEX FOR NAME CHECK
                         56    D5  0249    565          TSTL    R6                      ; ANY LOGICAL NAME FOR NEW PROCESS?
                         2B    13  024B    566          BEQL    140$                    ; NO, SKIP SEARCH
   58    00000000'FF47    D0  024D    567 120$:    MOVL    @SCH$GL_PCBVEC[R7],R8   ; FETCH A PCB ADDRESS
      00BE CA    00BE C8    B1  0255    568          CMPW    PCB$W_GRP(R8),PCB$W_GRP(R10)    ; SAME GROUP?
                         17    12  025C    569          BNEQ    130$                    ; NO, TRY ANOTHER
   70 A8    70 AA    56    29  025E    570          CMPC3   R6,PCB$T_LNAME(R10),PCB$T_LNAME(R8)    ; COMPARE NAMES
                         0F    12  0264    571          BNEQ    130$                    ; NO MATCH, CONTINUE
            50    0000'8F    3C  0266    572          MOVZWL  #SS$_DUPLNAM,R0         ; SET DUPLICATE NAME CODE
                     05D2    31  026B    573          BRW     ABORT                   ; AND ABORT CREATION
                             026E    574
                     0661    31  026E    575 125$:    BRW     ACCVIO                  ; GIVE ACCESS VIOLATION AND ABORT
                             0271    576 126$:    BUG_CHECK KRPEMPTY,FATAL        ; OUT-OF-LINE KRP ALLOCATION BUGCHECK
                  D5 57      F5  0275    577 130$:    SOBGTR  R7,120$                 ; CONTINUE SEARCH
                             0278    578 140$:    MOVSTRING       LIM=255,-       ; MOV IMAGE NAME TO BUFFER
                             0278    579                          SRC=IMAGE,-     ; FROM IMAGE STRING DESCRIPTOR
                             0278    580                          DST=IMAGE       ; TO PQB$T_IMAGE
                             027F    581          MOVSTRING       LIM=255,-       ; MOVE INPUT LOGICAL NAME TO BUFFER
                             027F    582                          SRC=INPUT,-     ; FROM INPUT STRING DESCRIPTOR
                             027F    583                          DST=INPUT       ; TO PQB$T_INPUT
                             0286    584          MOVSTRING       LIM=255,-       ; MOVE OUTPUT LOGICAL NAME TO BUFFER
                             0286    585                          SRC=OUTPUT,-    ; FROM OUTPUT STRING DESCRIPTOR
                             0286    586                          DST=OUTPUT      ; TO PQB$T_OUTPUT
                             028D    587          MOVSTRING       LIM=255,-       ; MOVE ERROR LOGICAL NAME TO BUFFER
                             028D    588                          SRC=ERROR,-     ; FROM ERROR STRING DESCRIPTOR
                             028D    589                          DST=ERROR       ; TO PQB$T_ERROR
                             0294    590
          46 A9      0F    90  0294    591          MOVB    #^B1111,PQB$B_MSGMASK(R9) ; DEFAULT MESSAGE FLAGS = ALL OF THEM
          03 5E      1F    E1  0298    592          BBC     #31,SP,1401$            ; IF SYSTEM SPACE STACK
                   00A6    31  029C    593          BRW     145$                    ; THEN NO P1 SPACE DATA BASE AVAILABLE
                     3F    BB  029F    594 1401$:   PUSHR   #^M<R0,R1,R2,R3,R4,R5>  ; SAVE MOVC REGISTERS
                             02A1    595
                             02A1    596 ;
                             02A1    597 ; TRANSLATE SYS$DISK USING LNM$SEARCH_ONE.
                             02A1    598 ; THIS TRANSLATES EXACTLY ONCE AND PASSES THE EQUIVALENCE STRING TO
                             02A1    599 ; THE CREATED PROCESS.  COMPLETELY BOGUS RESULTS WILL OCCUR IF THIS
                             02A1    600 ; TRANSLATION IS NOT EITHER A SYSTEM-WIDE LOGICAL NAME OR A PHYSICAL
                             02A1    601 ; DEVICE NAME.  THIS TOTALLY BOGUS IMPLEMENTATION MUST REMAIN THIS WAY
                             02A1    602 ; FOR COMPATIBILITY WITH PREVIOUS RELEASES.
                             02A1    603 ;
                             02A1    604 ; ALLOCATE A KERNEL REQUEST PACKET TO CONTAIN THE EQUIVALENCE STRING
                             02A1    605 ; (AND LOGICAL NAME WORK AREA) FOR LNM$SEARCH_ONE.
                             02A1    606
   57    00000000'GF      9E  02A1    607          MOVAB   G^CTL$GL_KRPFL,R7       ; GET KRP LISTHEAD
      56      04 B7      0F  02A8    608          REMQUE  @4(R7),R6               ; GET A KRP
                     C3    1D  02AC    609          BVS     126$                    ; BUG_CHECK IF ALL ARE IN USE
                     56    DD  02AE    610          PUSHL   R6                      ; SAVE POINTER TO KRP
   50    00000000'EF      3C  02B0    611          MOVZWL  EXE$GQ_SYSDISK,R0       ; GET DESCRIPTION OF "SYS$DISK" FOR CALL
   51    00000004'EF      D0  02B7    612          MOVL    EXE$GQ_SYSDISK+4,R1
      52    FD6E CF      3C  02BE    613          MOVZWL  LNM_TBL,R2              ; GET DESCRIPTION OF TABLE NAME
```

```
                53   FD6D CF    D0  02C3   614          MOVL    LNM_TBL+4,R3
                55    0103 8F   3C  02C8   615          MOVZWL  #LNM_ATTR,R5              ; CASE_BLIND AND USER_MODE
                 00000000'EF     16  02CD   616          JSB     LNM$SEARCH_ONE            ; TRANSLATE "SYS$DISK" (R4 HAS PCB)
                       09  50    E8  02D3   617          BLBS    R0,1405$                  ; SUCCESS!
                0000'8F    50    B1  02D6   618          CMPW    R0,#SS$_NOLOGNAM          ; NO TRANSLATION IS OKAY
                           15    12  02DB   619          BNEQ    141$                      ; ELSE ABORT
                           10    11  02DD   620          BRB     1406$                     ; GO RETURN KRP
                  50    04 A6    9A  02DF   621 1405$:   MOVZBL  LNMX$T_XLATION(R6),R0    ; GET SIZE OF TRANSLATION
                05C8 C9    50    90  02E3   622          MOVB    R0,PQB$T_DISK(R9)         ; STORE COUNT IN PQB
                  05 A6    50    28  02E8   623          MOVC3   R0,<LNMX$T_XLATION+1>(R6),-
                      05C9 C9          02EC   624                  <PQB$T_DISK+1>(R9)        ; COPY DEVICE NAME TO PQB
                      50   01    D0  02EF   625 1406$:   MOVL    #1,R0                     ; INDICATE SUCCESS
                      56   8E    D0  02F2   626 141$:    MOVL    (SP)+,R6                  ; RESTORE KRP POINTER
                   04 B7   66    0E  02F5   627          INSQUE  (R6),@4(R7)               ; RETURN KRP
                       59  50    E9  02F9   628          BLBC    R0,146$                   ; ABORT IF ERROR
                                     02FC   629
                                     02FC   630 ; MOVE MINIMUM AND MAXIMUM AUTHORIZED SECURITY CLEARANCE RECORDS FROM
                                     02FC   631 ; CREATOR'S PHD INTO THE PQB. THE FOLLOWING ASSUME STATEMENTS GUARANTEE THAT
                                     02FC   632 ; WE CAN SAFELY PERFORM THIS WITH A SINGLE MOVC3 INSTRUCTION.
                                     02FC   633
                                     02FC   634          ASSUME PQB$S_MIN_CLASS EQ PHD$S_MIN_CLASS
                                     02FC   635          ASSUME PQB$S_MAX_CLASS EQ PHD$S_MAX_CLASS
                                     02FC   636          ASSUME PQB$R_MAX_CLASS EQ <PQB$R_MIN_CLASS + PQB$S_MIN_CLASS>
                                     02FC   637          ASSUME PHD$R_MAX_CLASS EQ <PHD$R_MIN_CLASS + PHD$S_MIN_CLASS>
                                     02FC   638
          55    00000000'GF    D0  02FC   639          MOVL    G^CTL$GL_PHD,R5           ; GET CREATOR'S PHD ADDRESS
                       28    28  0303   640          MOVC3   #<PHD$S_MIN_CLASS+PHD$S_MAX_CLASS>,-
                      0114 C5        0305   641                  PHD$R_MIN_CLASS(R5),-
                        50 A9        0308   642                  PQB$R_MIN_CLASS(R9)
                                     030A   643
                      0100 8F   28   030A   644          MOVC3   #PQB$S_DDSTRING,-
        06C8 C9     00000000'9F      030E   645                  @#PIO$GT_DDSTRING,PQB$T_DDSTRING(R9) ; DEFAULT DIRECTORY
                                     0316   646
                                     0316   647 ; Move CLI and CLI table information from P1 space in one fell swoop:
                                     0316   648 ;       CTL$GT_CLINAME    -> PQB$T_CLI_NAME
                                     0316   649 ;       CTL$GT_TABLENAME  -> PQB$T_CLI_TABLE
                                     0316   650 ;       CTL$GT_SPAWNCLI   -> PQB$T_SPAWN_CLI            Optional
                                     0316   651 ;       CTL$GT_SPAWNTABLE -> PQB$T_SPAWN_TABLE          Optional
                                     0316   652
                                     0316   653          ASSUME PQB$T_CLI_TABLE   EQ <PQB$T_CLI_NAME  + PQB$S_CLI_NAME>
                                     0316   654          ASSUME PQB$T_SPAWN_CLI   EQ <PQB$T_CLI_TABLE + PQB$S_CLI_TABLE>
                                     0316   655          ASSUME PQB$T_SPAWN_TABLE EQ <PQB$T_SPAWN_CLI + PQB$S_SPAWN_CLI>
                                     0316   656
                                  3C 0316   657          MOVZWL  #<PQB$S_CLI_NAME+-        ; ASSUME WE'LL MOVE ALL FIELDS
                                     0317   658                    PQB$S_CLI_TABLE+-
                                     0317   659                    PQB$S_SPAWN_CLI+-
                                     0317   660                    PQB$S_SPAWN_TABLE>,-
                      50   0240 8F        0317   661                  R0
                   05 30 AC   0C   E0  031B   662          BBS     #PRC$V_CLISPEC,STSFLG(AP),143$ ; PASSING ON CLI SPECIFICATION?
                                  3C 0320   663          MOVZWL  #<PQB$S_CLI_NAME+-        ; NO, ONLY MOVE FIRST TWO FIELDS
                                     0321   664                    PQB$S_CLI_TABLE>,-
                      50   0120 8F        0321   665                  R0
                                  50 2C   0325   666 143$:   MOVC5   R0,-                   ; MOVE AS MANY FIELDS AS SPECIFIED
                                     0327   667                  @#CTL$GT_CLINAME,-        ;  FROM P1 SPACE
                00    00000000'9F      0327   668                  #0,-                      ;   ZEROING OPTIONAL FIELDS AS NEEDED
                                     032D   669                  #<PQB$S_CLI_NAME+-        ;    TO
                                     032D   670                    PQB$S_CLI_TABLE+-       ;     THE
```

```
                                032D   671                     PQB$S_SPAWN_CLI+-       ;       FULL
                                C32D   672                     PQB$S_SPAWN_TABLE>,-    ;        ALLOCATION
          0088 C9    0240 8F         032D   673                 PQB$T_CLI_NAME(R9)      ;         IN THE PQB
                                0333   674
                        3F      BA      0333   675         POPR    #^M<R0,R1,R2,R3,R4,R5>  ; RESTORE MOVC REGISTERS
      46 A9    00000000'9F     90      0335   676         MOVB    @#CTL$GB_MSGMASK,PQB$B_MSGMASK(R9) ; USE CREATOR'S MESSAGE FLAGS
      48 A9    00000000'9F     D0      033D   677         MOVL    @#CTL$GL_UAF_FLAGS,PQB$L_UAF_FLAGS(R9) ; AND FLAGS FROM UAF RECORD
                                0345   678 ;
                                0345   679 ;       ESTABLISH STATUS FLAG SETTINGS FOR PROCESS
                                0345   680 ;
            58    30 AC     D0      0345   681 145$:    MOVL    STSFLG(AP),R8           ; GET STATUS FLAG ARGUMENT
      50    58    12  0E    EF      0349   682         EXTZV   #STSFLGCNT,#<32-STSFLGCNT>,R8,R0 ; TEST MBZ FIELD
                        08    13      034E   683         BEQL    150$                    ; CORRECT IF ZERO
            50    0000'8F   3C      0350   684         MOVZWL  #SS$_IVSTSFLG,R0        ; ERROR, INVALID STATUS FLAG ARG
                      04ER    31      0355   685 146$:    BRW     ABORT                   ; ABORT CREATION
                                0358   686
          4C A9     58      D0      0358   687 150$:    MOVL    R8,PQB$L_CREPRC_FLAGS(R9) ; STORE FLAGS IN PQB
                            57      D4      035C   688         CLRL    R7              ; INITIALIZE INDEX FOR SCAN
      57    58      0E      57      EA      035E   689 160$:    FFS     R7,#STSFLGCNT,R8,R7     ; FIND AN ACTIVE STATUS FLAG
                            35      13      0363   690         BEQL    190$            ; NONE, FINISHED WITH SCAN
        51    FC96 CF47     3E      0365   691         MOVAW   STSFLGTBL[R7],R1        ; POINT TO TRANSLATION ENTRY
                  50        81      98      036B   692         CVTBL   (R1)+,R0        ; GET PRIVILEGE BIT NUMBER  TO CHECK
                            0F      18      036E   693         BGEQ    165$            ; NEGATIVE MEANS POSSIBLY NOT PRIVILEGED
            50    FF      8F      91      0370   694         CMPB    #-1,R0
                            16      13      0374   695         BEQL    170$            ; NO PRIVILEGE REQUIRED
                            5B      D5      0376   696         TSTL    R11
                            12      13      0378   697         BEQL    170$            ; NO PRIVILEGE REQUIRED FOR SUBPROCESS
      50    50      07      00      EF      037A   698         EXTZV   #0,#7,R0,R0       ; GET THE PRIVILEGE BIT NUMBER ONLY
          08 6C B4      50      E0      037F   699 165$:    BBS     R0,@PCB$L_PHD(R4),170$  ; THE CREATOR PROCESS MUST HAVE PRIVILEGE
            50    0000'8F   3C      0384   700         MOVZWL  #SS$_NOPRIV,R0          ; INDICATE NO PRIVILEGE ERROR
                      04B4    31      0389   701         BRW     ABORT                   ; AND ABORT PROCESS CREATION
                                038C   702
                  50    61      98      038C   703 170$:    CVTBL   (R1),R0                 ; GET BIT NUMBER IN STS
                            05      19      038F   704         BLSS    180$            ; NOT NEEDED IN PCB
            00 24 AA      50      E2      0391   705         BBSS    R0,PCB$L_STS(R10),180$  ; SET STSFLG IN NEW PCB
                            57      D6      0396   706 180$:    INCL    R7              ; NEXT BIT
                            C4      11      0398   707         BRB     160$            ; CONTINUE SCAN
                                039A   708 190$:                                ; DONE WITH STSFLG
                                039A   709 ;
                                039A   710 ; MISC PQB FLAGS
                                039A   711 ;
                  44 A9     B4      039A   712         CLRW    PQB$W_FLAGS(R9)         ; INITIALIZE THE FLAGS
            04 30 AC      0B      E1      039D   713         BBC     #PRC$V_IMGDMP,STSFLG(AP),195$
               44 A9      01      A8      03A2   714         BISW    #PQB$M_IMGDMP,PQB$W_FLAGS(R9) ; REQUEST IMAGE DUMP
                                03A6   715 195$:
                                03A6   716 ;
                                03A6   717 ; CHECK FOR MAXIMUM ALLOWED DETACHED PROCESSES
                                03A6   718 ;
                            5B      D5      03A6   719         TSTL    R11
                            16      13      03A8   720         BEQL    200$            ; NOT DETACHED
                         02204000   03AA   721         PCB_IBN = <1@PCB$V_INTER>+<1@PCB$V_BATCH>+<1@PCB$V_NETWRK>
      24 AA    02204000  8F      D3      03AA   722         BITL    #PCB_IBN,PCB$L_STS(R10)
                            0C      12      03B2   723         BNEQ    200$            ; NO CHECK ON NETWORK, BATCH, OR INTERACTIVE
            55    0080 CA     D0      03B4   724         MOVL    PCB$L_JIB(R10),R5       ; GET THE JIB OF NEW PROCESS
            57    0080 C4     D0      03B9   725         MOVL    PCB$L_JIB(R4),R7        ; GET THE JIB OF PARENT PROCESS
                            03      12      03BE   726         BNEQ    205$            ; IF NO JIB - WE ARE BOOTING
                        008D    31      03C0   727 200$:    BRW     240$            ; BRANCH AID
```

```
                                03C3    728
                    52 A7   B0  03C3    729 205$:   MOVW    JIB$W_MAXDETACH(R7),-   ; DETACHED PROCESS LIMIT
                    52 A5       03C6    730                 JIB$W_MAXDETACH(R5)     ; PROPAGATE THE LIMITS
                    50 A7   B0  03C8    731         MOVW    JIB$W_MAXJOBS(R7),-     ; MAX PROCESS LIMIT
                    50 A5       03CB    732                 JIB$W_MAXJOBS(R5)
          55  50 A7    52 A7   A9  03CD    733         BISW3   JIB$W_MAXDETACH(R7),JIB$W_MAXJOBS(R7),R5
                       7B   13  03D3    734         BEQL    240$                    ; NO LIMIT
                       56   D4  03D5    735         CLRL    R6
                       58   D4  03D7    736         CLRL    R8
                  0E1F 8F   BB  03D9    737         PUSHR   #^M<R0,R1,R2,R3,R4,R9,R10,R11>
                  59   02   D0  03DD    738         MOVL    #2,R9                   ; PROCESS INDEX - AFTER NULL AND SWAPPER
                                03E0    739         SETIPL  IPL_SYNCH
          5A  00000000'FF49   D0  03E7    740 210$:   MOVL    @SCH$GL_PCBVEC[R9],R10  ; GET A PCB
            00000000'FF   5A  D1  03EF    741         CMPL    R10,@SCH$GL_PCBVEC      ; IS IT NULL PROCESS PCB
                       25   13  03F6    742         BEQL    220$                    ; YES
                    1C AA   D5  03F8    743         TSTL    PCB$L_OWNER(R10)        ; IS IT SUBPROCESS
                       20   12  03FB    744         BNEQ    220$
                 1B 24 AA   15  E0  03FD    745         BBS     #PCB$V_NETWRK,PCB$L_STS(R10),220$ ; DON'T COUNT NETWORK JOBS
                   5B   0080 CA  D0  0402    746         MOVL    PCB$L_JIB(R10),R11      ; JIB OF PROCESS BEING CHECKED
          0C AB    0C A7   0C  29  0407    747         CMPC3   #JIB$S_USERNAME,JIB$T_USERNAME(R7),JIB$T_USERNAME(R11)
                       0E   12  040D    748         BNEQ    220$                    ; NOT THE SAME USER
                       58   D6  040F    749         INCL    R8                      ; ONE MORE TOTAL JOB
                     02004000  0411    750         PCB_IB = <1@PCB$V_INTER>+<1@PCB$V_BATCH>
       24 AA    02004000 8F   D3  0411    751         BITL    #PCB_IB,PCB$L_STS(R10)
                       02   12  0419    752         BNEQ    220$                    ; INTERACTIVE OR BATCH
                       56   D6  041B    753         INCL    R6                      ; ONE MORE TO COUNT AGAINST DETACHED
       C2 59    00000000'EF   F3  041D    754 220$:   AOBLEQ  SCH$GL_MAXPIX,R9,210$
                  0E1F 8F   BA  0425    755         POPR    #^M<R0,R1,R2,R3,R4,R9,R10,R11>
                                0429    756         SETIPL  #0
                55   50 A7   3C  042C    757         MOVZWL  JIB$W_MAXJOBS(R7),R5
                       05   13  0430    758         BEQL    225$                    ; NO LIMIT
                   58   55   C2  0432    759         SUBL    R5,R8
                       0B   1E  0435    760         BGEQU   230$                    ; OVER LIMIT (INCLUDING THIS PROCESS)
              55   52 A7   3C  0437    761 225$:   MOVZWL  JIB$W_MAXDETACH(R7),R5
                       13   13  043B    762         BEQL    240$                    ; NO LIMIT
                   56   55   C2  043D    763         SUBL    R5,R6
                       0E   1F  0440    764         BLSSU   240$                    ; OVER LIMIT (INCLUDING THIS PROCESS)
    50    00000000'8F   D0  0442    765 230$:   MOVL    #SS$_EXPRCLM,R0
                  03F4   31  0449    766         BRW     ABORT
                                044C    767 IPL_SYNCH:
                  00000008  044C    768         .LONG   IPL$_SYNCH
                            0450    769 ;
                            0450    770 240$:
                            0450    771 ;
                            0450    772         .DSABL  LSB
```

```
                     0450   774          .SBTTL  ESTABLISH QUOTAS FOR NEW PROCESS
                     0450   775 ;-----------------------------------------------------------
                     0450   776 ;
                     0450   777 ;       PROCESS QUOTA BLOCK
                     0450   778 ;
                     0450   779 ;-----------------------------------------------------------
                     0450   780
                     0450   781 ;
                     0450   782 ;       THE PROCESS QUOTA LIST, IF SUPPLIED, HAS THE FOLLOWING STRUCTURE
                     0450   783 ;       EACH QUOTA IS INTRODUCED AND IDENTIFIED BY A CODE BYTE, PQL$_?????,
                     0450   784 ;       WHICH IS FOLLOWED BY A LONGWORD CONTAINING THE QUOTA VALUE.
                     0450   785 ;
                     0450   786 ;       THE QUOTA VALUES SUPPLIED ARE MAXIMIZED WITH THE REQUIRED MINIMUM
                     0450   787 ;       VALUES AND REPLACE THE DEFAULT VALUE FOR EACH SPECIFIED QUOTA.
                     0450   788 ;       ONLY IF THE PROCESS CREATION IS SUCCESSFUL ARE THE DEDUCTIBLE
                     0450   789 ;       QUOTAS SUBTRACTED FROM THOSE OF A PROCESS CREATING A DETACHED PROCESS.
                     0450   790 ;
                     0450   791 ;       IF DUPLICATE QUOTA ENTRIES ARE FOUND, THE LAST ONE ENCOUNTERED IS
                     0450   792 ;       THE ONE THAT IS USED.
                     0450   793 ;
                     0450   794
                     0450   795 QUOTALIST:                                        ;
                1F   BB   0450   796          PUSHR   #^M<R0,R1,R2,R3,R4>       ; SAVE REGISTERS FOR MOVC
                38   28   0452   797          MOVC3   #<<PQL$_LENGTH-1>*4>,-    ; COPY DEFAULTS TO PQB
      00000004'EF         0454   798                  PQL$AL_DEFAULT+4,-        ; AS ASSUMED VALUES FOR
            0C A9         0459   799                  PQB$L_ASTLM(R9)           ; QUOTAS
                1F   BA   045B   800          POPR    #^M<R0,R1,R2,R3,R4>       ; RESTORE REGISTERS
         55   6C A4   D0   045D   801          MOVL    PCB$L_PHD(R4),R5          ; FOR SWAPPER POINT TO REAL PHD SINCE NO
00000000'EF   60 A4   D1   0461   802          CMPL    PCB$L_PID(R4),SCH$GL_SWPPID     ; IS IT THE SWAPPER
                07   13   0469   803          BEQL    5$                        ; YES, USE PCB POINTER TO PHD
   55   00000000'9F   D0   046B   804          MOVL    @#CTL$GL_PHD,R5           ; GET POINTER TO PHD WINDOW IN
                     0472   805                                                   ; CONTROL REGION FOR WINDOW
   57   1C AC   D0   0472   806 5$:     MOVL    QUOTA(AP),R7                      ; GET POINTER TO QUOTA LIST
                28   13   0476   807          BEQL    NOQLIST                   ; NONE SPECIFIED
                     0478   808 10$:     IFNORD  #1,(R7),30$                       ; CHECK FOR ACCESSIBILITY
      56   87   9A   047E   809          MOVZBL  (R7)+,R6                  ; GET CODE
                     0481   810          ASSUME  PQL$_LISTEND EQ 0         ;
                1D   13   0481   811          BEQL    NOQLIST                   ; DONE IF PQL$_ENDLIST
                     0483   812          IFNORD  #4,(R7),30$                       ; CHECK QUOTA FOR ACCESSIBILITY
      0F   56   B1   0489   813          CMPW    R6,#PQL$_LENGTH           ; CHECK FOR LEGAL QUOTA NUMBER
                07   18   048C   814          BGEQ    20$                       ; INVALID IF GEQ
   08 A946   87   D0   048E   815          MOVL    (R7)+,PQB$L_ASTLM-4(R9)[R6] ; MERGE INTO PQB QUOTA LIST
                E3   11   0493   816          BRB     10$                       ; GO GET NEXT QUOTA SPECIFIED
   50   0000'8F   3C   0495   817 20$:     MOVZWL  #SS$_IVQUOTAL,R0          ; INVALID QUOTA LIST
          03A3   31   049A   818          BRW     ABORT                     ; SIGNAL ERROR CONDITION AND ABORT CREATE
                     049D   819
          0432   31   049D   820 30$:     BRW     ACCVIO                    ; ABORT WITH ACCESS VIOLATION
                     04A0   821
                     04A0   822 NOQLIST:                                       ; DONE MERGING QUOTAS SPECIFIED
   56   0000003C'EF   DE   04A0   823          MOVAL   PQL$AL_MIN+<4*PQL$_LENGTH>,R6   ; SET POINTER TO BASE OF MIN VALUES
         53   44 A9   DE   04A7   824          MOVAL   <PQB$L_ASTLM+<4*<PQL$_LENGTH-1>>>(R9),R3 ; SET BASE OF QUOTA VALUES
            58   0E   3C   04AB   825          MOVZWL  #<PQL$_LENGTH-1>,R8       ; SET COUNT FOR SCAN
   57   0080 CA   D0   04AE   826          MOVL    PCB$L_JIB(R10),R7         ; GET ADDRESS OF JOB INFORMATION BLOCK
                52   D4   04B3   827          CLRL    R2                        ; INDICATE UNRESTRICTED QUOTAS ALLOWED
                5B   D5   04B5   828          TSTL    R11                       ; IS IT A SUBPROCESS
                0C   13   04B7   829          BEQL    5$                        ; YES
                     04B9   830          IFPRIV  DETACH,10$                        ; UNRESTRICTED IS OK
```

```
                     04BF    831          IFPRIV  CMKRNL,10$
52   0080 C4   D0   04C5    832 5$:      MOVL    PCB$L_JIB(R4),R2        ; INDICATE RESTRICTED QUOTAS (JIB ADDRESS)
76   73        D1   04CA    833 10$:     CMPL    -(R3),-(R6)             ; CHECK AGAINST MINIMUM ALLOWABLE VALUE
          03   18   04CD    834          BGEQ    20$                     ; BR IF ABOVE MINIMUM
63   66        D0   04CF    835          MOVL    (R6),(R3)               ; FORCE TO MINIMUM
                     04D2    836 20$:
50   63        D0   04D2    837          MOVL    (R3),R0                 ; GET QUOTA REQUEST VALUE
          06   10   04D5    838          BSBB    40$                     ; PROCESS QUOTA
     F0 58     F5   04D7    839          SOBGTR  R8,10$                  ; LOOP FOR ALL QUOTAS
      01A8     31   04DA    840          BRW     ITEMLIST                ; GO PROCESS THE ITEM LIST
                     04DD    841
                     04DD    842 40$:     CASE    R8,LIMIT=#1,<-          ; SWITCH ON TYPE OF QUOTA
                     04DD    843                  QASTLM,-                ; 1 => AST LIMIT
                     04DD    844                  QBIOLM,-                ; 2 => BUFFERED I/O LIMIT
                     04DD    845                  QBYTLM,-                ; 3 => BUFIO BYTE COUNT LIMIT
                     04DD    846                  QCPULM,-                ; 4 => CPU TIME LIMIT
                     04DD    847                  QDIOLM,-                ; 5 => DIRECT I/O LIMIT
                     04DD    848                  QFILLM,-                ; 6 => OPEN FILE LIMIT
                     04DD    849                  QPGFLQUOTA,-            ; 7 => PAGING FILE QUOTA
                     04DD    850                  QPRCLM,-                ; 8 => SUB-PROCESS LIMIT
                     04DD    851                  QTQELM,-                ; 9 => TIMER QUEUE ENTRY LIMIT
                     04DD    852                  QWSQUOTA,-              ;10 => WORKING SET QUOTA
                     04DD    853                  QWSDEFAULT,-            ;11 => WORKING SET DEFAULT
                     04DD    854                  QENQLM,-                ;12 => ENQUEUE LIMIT
                     04DD    855                  QWSEXTENT,-             ;13 => WORKING SET EXTENT
                     04DD    856                  QJTQUOTA-               ;14 => JOB-WIDE LOGICAL NAME TABLE QUOTA
                     04DD    857                  >;
                     04FD    858
                     04FD    859 QJTQUOTA:                                ; NO SPECIAL PROCESSING FOR JTQUOTA
          05         04FD    860          RSB                             ;
                     04FE    861
                     04FE    862 QASTLM:                                  ; AST LIMIT
40 A5     50   B1   04FE    863          CMPW    R0,PHD$W_ASTLM(R5)      ; CHECK FOR IN LIMIT
          08   1B   0502    864          BLEQU   10$                     ; YES, CONTINUE
          52   D5   0504    865          TSTL    R2                      ; UNRESTRICTED DETACHED CREATE?
          04   13   0506    866          BEQL    10$                     ; YES, ALLOW ANYTHING
50   40   A5   B0   0508    867          MOVW    PHD$W_ASTLM(R5),R0      ; NO, LIMIT TO MAXIMUM
38 AA     50   B0   050C    868 10$:     MOVW    R0,PCB$W_ASTCNT(R10)    ; SET AS WORKING AST COUNT
          05   10   0510    869          RSB                             ; NEXT QUOTA
                     0511    870
                     0511    871 QBIOLM:                                  ; BUFFERED I/O LIMIT
3C A4     50   B1   0511    872          CMPW    R0,PCB$W_BIOLM(R4)      ; CHECK FOR IN LIMIT
          08   1B   0515    873          BLEQU   10$                     ; YES, CONTINUE
          52   D5   0517    874          TSTL    R2                      ; UNRESTRICTED DETACHED CREATE?
          04   13   0519    875          BEQL    10$                     ; YES ALLOW ANYTHING
50   3C   A4   B0   051B    876          MOVW    PCB$W_BIOLM(R4),R0      ; NO, LIMIT TO CURRENT VALUE
3C AA     50   B0   051F    877 10$:     MOVW    R0,PCB$W_BIOLM(R10)     ; SET LIMIT
3A AA     50   B0   0523    878          MOVW    R0,PCB$W_BIOCNT(R10)    ; AND WORKING COUNT
          05   10   0527    879          RSB                             ; NEXT QUOTA
                     0528    880
                     0528    881 QBYTLM:                                  ; BUFFERED I/O BYTE LIMIT
          5B   D5   0528    882          TSTL    R11                     ; DETACHED CREATE?
          1A   13   052A    883          BEQL    10$                     ; BR IF NOT
          52   D5   052C    884          TSTL    R2                      ; UNRESTRICTED QUOTAS?
          0A   13   052E    885          BEQL    5$                      ; YES
24 A2     50   D1   0530    886          CMPL    R0,JIB$L_BYTLM(R2)      ; IS IT WITHIN LIMITS
          04   1B   0534    887          BLEQU   5$                      ; YES
```

```
                50   24 A2    D0  0536   888                MOVL    JIB$L_BYTLM(R2),R0      ; LIMIT TO CURRENT VALUE
                24 A7    50    D0  053A   889 5$:            MOVL    R0,JIB$L_BYTLM(R7)      ; SET IN JIB OF MASTER PROCESS
                20 A7    50    D0  053E   890                MOVL    R0,JIB$L_BYTCNT(R7)     ; BOTH LIMIT AND COUNT VALUES
                6C A7    50    D0  0542   891                MOVL    R0,JIB$L_ORG_BYTLM(R7)  ; AND ORIGINAL VALUE FIELD
                         05    0546   892 10$:               RSB                             ; NEXT QUOTA
                               0547   893
                               0547   894 QCPULM:                                          ; CPU TIME LIMIT
                      52       D5  0547   895                TSTL    R2                      ; UNRESTRICTED DETACHED CREATE?
                      34       13  0549   896                BEQL    20$                     ; YES, ALLOW ANYTHING
                      5B       D5  054B   897                TSTL    R11                     ; IS IT A DETACHED PROCESS
                      0C       13  054D   898                BEQL    3$                      ; NO
                5C A5    50    D1  054F   899                CMPL    R0,PHD$L_CPULIM(R5)     ; IS IT WITHIN LIMITS
                      2A       1B  0553   900                BLEQU   20$                     ; YES
                50   5C A5    D0  0555   901                MOVL    PHD$L_CPULIM(R5),R0     ; LIMIT IT TO CURRENT VALUE
                      24       11  0559   902                BRB     20$
                51   5C A5    D0  055B   903 3$:            MOVL    PHD$L_CPULIM(R5),R1     ; GET CURRENT PROCESS LIMIT
                      1E       13  055F   904                BEQL    20$                     ; NO LIMIT ON CURRENT PROCESS
                51   38 A5    C2  0561   905                SUBL    PHD$L_CPUTIM(R5),R1     ; DETERMINE REMAINING LIMIT
                      15       1B  0565   906                BLEQU   10$                     ; BRANCH IF WILL EXCEED LIMIT
                      50       D5  0567   907                TSTL    R0                      ; IS REQUESTED VALUE INFINITY?
                      0C       12  0569   908                BNEQ    5$                      ; NO, GO SEE IF REQUEST CAN BE FILLED
                      51       D6  056B   909                INCL    R1                      ; IF REQUESTING INFINITY, GIVE
                   51    01    CA  056D   910                BICL2   #1,R1                   ;  HALF OF CALLER'S LIMIT BY CLEARING
             50    51    1F    9C  0570   911                ROTL    #31,R1,R0               ;  LOW BIT AND ROTATING THAT TO SIGN BIT
                   63    50    D0  0574   912                MOVL    R0,(R3)                 ; STORE NEW LIMIT IN PQB
                   51    50    D1  0577   913 5$:            CMPL    R0,R1                   ; CHECK FOR IN LIMIT
                         03    1F  057A   914                BLSSU   20$                     ; YES, CONTINUE
                      00F9     30  057C   915 10$:           BSBW    OVERCHECK               ; CHECK FOR LEGAL TO EXCEED
                         05    057F   916 20$:           RSB                             ; NEXT QUOTA
                               0580   917
                               0580   918 QDIOLM:                                         ; DIRECT I/O LIMIT
             40 A4    50    B1  0580   919                CMPW    R0,PCB$W_DIOLM(R4)      ; CHECK FOR IN LIMIT
                      08       1B  0584   920                BLEQU   10$                     ; YES, CONTINUE
                      52       D5  0586   921                TSTL    R2                      ; UNRESTRICTED DETACHED CREATE?
                      04       13  0588   922                BEQL    10$                     ; YES, ALLOW ANYTHING
                50   40 A4    B0  058A   923                MOVW    PCB$W_DIOLM(R4),R0      ; ELSE LIMIT TO CURRENT VALUE
             40 AA    50    B0  058E   924 10$:           MOVW    R0,PCB$W_DIOLM(R10)     ; SET LIMIT
             3E AA    50    B0  0592   925                MOVW    R0,PCB$W_DIOCNT(R10)    ; AND SET WORKING COUNT
                         05    0596   926                RSB                             ;
                               0597   927
                               0597   928 QFILLM:                                         ; OPEN FILE LIMIT
                      5B       D5  0597   929                TSTL    R11                     ; IS THIS A DETACHED CREATE?
                      16       13  0599   930                BEQL    10$                     ; BR IF NOT
                      52       D5  059B   931                TSTL    R2                      ; UNRESTRICTED QUOTAS?
                      0A       13  059D   932                BEQL    5$                      ; YES
             32 A2    50    B1  059F   933                CMPW    R0,JIB$W_FILLM(R2)      ; IS IT WITHIN LIMITS
                      04       1B  05A3   934                BLEQU   5$                      ; YES
                50   32 A2    B0  05A5   935                MOVW    JIB$W_FILLM(R2),R0      ; LIMIT TO CURRENT VALUE
             30 A7    50    B0  05A9   936 5$:            MOVW    R0,JIB$W_FILCNT(R7)     ; SET FILE COUNT FOR MASTER PROCESS
             32 A7    50    B0  05AD   937                MOVW    R0,JIB$W_FILLM(R7)      ; AND LIMIT VALUE
                         05    05B1   938 10$:           RSB                             ; NEXT QUOTA
                               05B2   939
                               05B2   940 QPGFLQUOTA:                                     ; PAGE FILE QUOTA
                      5B       D5  05B2   941                TSTL    R11                     ; DETACHED CREATE?
                      16       13  05B4   942                BEQL    10$                     ; BR IF NOT
                      52       D5  05B6   943                TSTL    R2                      ; UNRESTRICTED QUOTAS?
                      0A       13  05B8   944                BEQL    5$                      ; YES
```

```
                38 A2    50    D1   05BA   945           CMPL    R0,JIB$L_PGFLQUOTA(R2)  ; IS IT WITHIN LIMITS
                         04    1B   05BE   946           BLEQU   5$                      ; YES
                50   38 A2    D0   05C0   947           MOVL    JIB$L_PGFLQUOTA(R2),R0  ; LIMIT TO CURRENT VALUE
                38 A7    50    D0   05C4   948 5$:       MOVL    R0,JIB$L_PGFLQUOTA(R7)  ; SET PAGE FILE QUOTA
                3C A7    50    D0   05C8   949           MOVL    R0,JIB$L_PGFLCNT(R7)    ; AND COUNT
                               05   05CC   950 10$:      RSB                             ; NEXT QUOTA
                                    05CD   951
                                    05CD   952 QPRCLM:                                  ; SUBPROCESS QUOTA
                         5B    D5   05CD   953           TSTL    R11                     ; IS THIS A DETACHED CREATE?
                         12    13   05CF   954           BEQL    10$                     ; BR IF NOT
                         52    D5   05D1   955           TSTL    R2                      ; UNRESTRICTED QUOTAS?
                         0A    13   05D3   956           BEQL    5$                      ; YES
                46 A2    50    B1   05D5   957           CMPW    R0,JIB$W_PRCLIM(R2)     ; IS IT WITHIN LIMITS
                         04    1B   05D9   958           BLEQU   5$                      ; YES
                50   46 A2    B0   05DB   959           MOVW    JIB$W_PRCLIM(R2),R0     ; LIMIT TO CURRENT VALUE
                46 A7    50    B0   05DF   960 5$:       MOVW    R0,JIB$W_PRCLIM(R7)     ; AND LIMIT
                               05   05E3   961 10$:      RSB                             ; NEXT QUOTA
                                    05E4   962
                                    05E4   963 QTQELM:                                  ; TIMER QUEUE ENTRY QUOTA
                         5B    D5   05E4   964           TSTL    R11                     ; IS THIS A DETACHED CREATE?
                         16    13   05E6   965           BEQL    10$                     ; BR IF NOT
                         52    D5   05E8   966           TSTL    R2                      ; UNRESTRICTED QUOTAS?
                         0A    13   05EA   967           BEQL    5$                      ; YES
                34 A2    50    B1   05EC   968           CMPW    R0,JIB$W_TQCNT(R2)      ; IS IT WITHIN LIMITS
                         04    1B   05F0   969           BLEQU   5$                      ; YES
                50   34 A2    B0   05F2   970           MOVW    JIB$W_TQCNT(R2),R0      ; LIMIT TO CURRENT VALUE
                34 A7    50    B0   05F6   971 5$:       MOVW    R0,JIB$W_TQCNT(R7)      ; SET TIMER QUEUE QUOTA FOR JOB
                36 A7    50    B0   05FA   972           MOVW    R0,JIB$W_TQLM(R7)       ; AND LIMIT VALUE
                               05   05FE   973 10$:      RSB                             ; NEXT QUOTA
                                    05FF   974
                                    05FF   975 QWSQUOTA:                                ; WORKING SET QUOTA
                      48    10   05FF   976           BSBB    MAXWSCNT                ; LIMIT TO MAX WORKING SET COUNT
                50   08 A5    A0   0601   977           ADDW    PHD$W_WSLIST(R5),R0     ; ADD BASE OF WORKING SET LIST
                18 A5    50    B1   0605   978           CMPW    R0,PHD$W_WSQUOTA(R5)    ; CHECK FOR IN LIMIT
                         09    1F   0609   979           BLSSU   10$                     ; YES,CONTINUE
                         52    D5   060B   980           TSTL    R2                      ; UNRESTRICTED DETACHED PROCESS?
                         05    13   060D   981           BEQL    10$                     ; YES, USE AS IS
                50   18 A5    01    A1   060F   982           ADDW3   #1,PHD$W_WSQUOTA(R5),R0 ; ELSE FORCE TO CORRECT MAXIMUM
                63   50   08 A5    A3   0614   983 10$:      SUBW3   PHD$W_WSLIST(R5),R0,(R3); REMOVE BIAS
                               05   0619   984           RSB                             ; NEXT QUOTA
                                    061A   985
                                    061A   986 QWSEXTENT:                               ; WORKING SET EXTENT
                      30    10   061A   987           BSBB    MAXWSCNT                ; LIMIT TO MAX WORKING SET COUNT
                30 A9    50    D1   061C   988           CMPL    R0,PQB$L_WSQUOTA(R9)    ; CHECK AGAINST QUOTA
                         04    1E   0620   989           BGEQU   5$                      ; MORE, USE AS IS
                50   30 A9    D0   0622   990           MOVL    PQB$L_WSQUOTA(R9),R0    ; MAXIMIZE WITH WSQUOTA
                50   08 A5    A0   0626   991 5$:       ADDW    PHD$W_WSLIST(R5),R0     ; ADD BASE OF WORKING SET LIST
                16 A5    50    B1   062A   992           CMPW    R0,PHD$W_WSEXTENT(R5)   ; CHECK FOR IN LIMIT
                         09    1F   062E   993           BLSSU   10$                     ; YES,CONTINUE
                         52    D5   0630   994           TSTL    R2                      ; UNRESTRICTED DETACHED PROCESS?
                         05    13   0632   995           BEQL    10$                     ; YES, USE AS IS
                50   16 A5    01    A1   0634   996           ADDW3   #1,PHD$W_WSEXTENT(R5),R0; ELSE FORCE TO CORRECT MAXIMUM
                63   50   08 A5    A3   0639   997 10$:      SUBW3   PHD$W_WSLIST(R5),R0,(R3); REMOVE BIAS
                               05   063E   998           RSB                             ; NEXT QUOTA
                                    063F   999
                                    063F  1000 QWSDEFAULT:                              ; WORKING SET LIST DEFAULT
                      0B    10   063F  1001           BSBB    MAXWSCNT                ; LIMIT TO MAXIMUM WORKING SET COUNT
```

```
                30 A9    50    D1  0641  1002          CMPL    R0,PQB$L_WSQUOTA(R9)    ; CHECK AGAINST QUOTA
                         04    1B  0645  1003          BLEQU   10$                     ; LESS, USE AS IS
                63   30 A9    D0  0647  1004          MOVL    PQB$L_WSQUOTA(R9),(R3)  ; MINIMIZE WITH WSQUOTA
                               05  064B  1005 10$:     RSB                             ; NEXT QUOTA
                                   064C  1006
                                   064C  1007
                                   064C  1008 MAXWSCNT:                               ; LIMIT WSQUOTA OR DEFAULT TO MAXIMUL SIZE
      00000000'EF    50    B1  064C  1009          CMPW    R0,SGN$GL_MAXWSCNT      ; COMPARE WITH MAXIMUM WS LIST LENGTH
                         07    1B  0653  1010          BLEQU   10$                     ; BR IF WITHIN LEGAL RANGE
50    00000000'EF    B0  0655  1011          MOVW    SGN$GL_MAXWSCNT,R0      ; FORCE TO LEGAL VALUE
                               05  065C  1012 10$:     RSB                             ;
                                   065D  1013
                                   065D  1014 QENQLM:                                 ; ENQUEUE QUOTA
                         5B    D5  065D  1015          TSTL    R11                     ; IS THIS A DETACHED CREATE?
                         16    13  065F  1016          BEQL    10$                     ; BR IF NOT
                         52    D5  0661  1017          TSTL    R2                      ; UNRESTRICTED QUOTAS?
                         0A    13  0663  1018          BEQL    5$                      ; YES
                4C A2    50    B1  0665  1019          CMPW    R0,JIB$W_ENQCNT(R2)     ; IS IT WITHIN LIMITS
                         04    1B  0669  1020          BLEQU   5$                      ; YES
                50   4C A2    B0  066B  1021          MOVW    JIB$W_ENQCNT(R2),R0     ; LIMIT TO CURRENT VALUE
                4C A7    50    B0  066F  1022 5$:      MOVW    R0,JIB$W_ENQCNT(R7)     ; SET ENQUEUE QUOTA FOR JOB
                4E A7    50    B0  0673  1023          MOVW    R0,JIB$W_ENQLM(R7)      ; AND LIMIT VALUE
                               05  0677  1024 10$:     RSB                             ; NEXT QUOTA
                                   0678  1025
```

```
                     0678  1027          .SBTTL  OVERCHECK - CHECK FOR LEGAL TO EXCEED QUOTA
                     0678  1028 ;++
                     0678  1029 ; FUNCTIONAL DESCRIPTION:
                     0678  1030 ;       OVERCHECK CHECKS TO SEE IF THE PROCESS BEING CREATED IS A DETATCHED
                     0678  1031 ;       PROCESS.   IF A DETATCHED PROCESS IS BEING CREATED, CONTROL RETURNS
                     0678  1032 ;       INLINE.  OTHERWISE THE CREATE IS ABORTED BY BRANCHING TO ABORT
                     0678  1033 ;       WITH THE STATUS CODE SS$_NOQUOTA.
                     0678  1034 ;
                     0678  1035 ; INPUT PARAMETERS:
                     0678  1036 ;       R4 - PCB ADDRESS OF CURRENT PROCESS
                     0678  1037 ;       R5 - ADDRESS OF PHD FOR CURRENT PROCESS (WINDOW IN P1 SPACE)
                     0678  1038 ;       R9 - ADDRESS OF PROCESS QUOTA BLOCK
                     0678  1039 ;       R10- ADDRESS OF PCB FOR NEW PROCESS
                     0678  1040 ;       R11- UIC FOR CREATED PROCESS (0 => SUBPROCESS)
                     0678  1041 ;
                     0678  1042 ;--
                     0678  1043
                     0678  1044 OVERCHECK:                                      ;
          5B   D5    0678  1045          TSTL    R11                           ; CHECK FOR SUBPROCESS CREATE
          08   12    067A  1046          BNEQ    10$                           ; YES, IGNORE OVER LIMIT
    50 0000'8F   3C  067C  1047          MOVZWL  #SS$_EXQUOTA,R0               ; SET ERROR STATUS CODE
        01BC   31    0681  1048          BRW     ABORT                         ; NO, ABORT CREATE
                     0684  1049
                     05  0684  1050 10$:    RSB                               ; RETURN
```

```
                     0685  1053          .SBTTL  PROCESS THE ITEM LIST
                     0685  1054 ;-----------------------------------------------------------------
                     0685  1055 ;
                     0685  1056 ;       PROCESS ITEM LIST
                     0685  1057 ;
                     0685  1058 ;-----------------------------------------------------------------
                     0685  1059
                     0685  1060 ;
                     0685  1061 ;       THE PROCESS ITEM LIST, IF SUPPLIED, HAS THE FOLLOWING STRUCTURE
                     0685  1062 ;       EACH ITEM HAS A 2 LONGWORD FIELD.  THE FIRST LONGWORD HAS TWO
                     0685  1063 ;       SUBFIELDS, A WORD OF ITEM LENGTH, FOLLOWED BY A CODE WORD PRC$_?????,
                     0685  1064 ;       WHICH IS FOLLOWED BY A LONGWORD CONTAINING THE ITEM VALUE.
                     0685  1065 ;
                     0685  1066 ;
                     0685  1067 ;       IF DUPLICATE ITEM LIST ENTRIES ARE FOUND, THE LAST ONE ENCOUNTERED IS
                     0685  1068 ;       THE ONE THAT IS USED.
                     0685  1069 ;
                     0685  1070 ;       NOTE THAT THE PAGE FILE INDEX AND CHARACTERISTICS WILL BE TREATED AS
                     0685  1071 ;       ADVICE ONLY.  IF THEY ARE INVALID OR CANNOT BE MET (NO FILE AVAILABLE),
                     0685  1072 ;       THE NORMAL ALGORITHM FOR ASSIGNING PAGE FILES WILL BE USED IN SHELL.
                     0685  1073 ;
                     0685  1074          ASSUME PRC$_PGFLCHAR EQ 1
                     0685  1075          ASSUME PRC$_PGFLINDEX EQ 2
                     0685  1076          ASSUME PRC$_INPUT_ATT EQ 3
                     0685  1077          ASSUME PRC$_OUTPUT_ATT EQ 4
                     0685  1078          ASSUME PRC$_ERROR_ATT EQ 5
                     0685  1079 ;
                     0685  1080 ITEMLIST:
          0D   6C    D1   0685  1081          CMPL    (AP),#ITMLST_ARG        ; WAS THE ITEM LIST ARGUMENT SUPPLIED?
               5D    1F   0688  1082          BLSSU   NO_ITMLST               ; NO
                          068A  1083          IFNORD  #4,ITMLST(AP),40$       ; CAN WE READ THE ITEMLIST POINTER
      57   34    AC    D0   0691  1084          MOVL    ITMLST(AP),R7           ; GET THE POINTER
               50    13   0695  1085          BEQL    NO_ITMLST               ; NONE SPECIFIED
                          0697  1086 10$:      IFNORD  #4,(R7),40$             ; CAN WE READ THE CODE AND LENGTH
           50   87    3C   069D  1087          MOVZWL  (R7)+,R0                ; GET LENGTH WORD
           56   87    3C   06A0  1088          MOVZWL  (R7)+,R6                ; GET THE CODE
                42    13   06A3  1089          BEQL    NO_ITMLST               ; END OF THE LIST
                          06A5  1090          IFNORD  #4,(R7),40$             ; CAN WE READ THE ITEM VALUE
                05    10   06AB  1091          BSBB    20$                     ; PROCESS THE ITEM
           57   04    C0   06AD  1092          ADDL    #4,R7
                E5    11   06B0  1093          BRB     10$                     ; NEXT
                          06B2  1094
                          06B2  1095 20$:      CASE    R6,LIMIT=#1,<-
                          06B2  1096                  ITM_PGFLCHAR,-
                          06B2  1097                  ITM_PGFLINDEX,-
                          06B2  1098                  ITM_INPUT_ATT,-
                          06B2  1099                  ITM_OUTPUT_ATT,-
                          06B2  1100                  ITM_ERROR_ATT,-
                          06B2  1101                  >
   50   00000000'8F    D0   06C0  1102          MOVL    #SS$_BADPARAM,R0
                0176    31   06C7  1103          BRW     ABORT
                          06CA  1104
                0205    31   06CA  1105 40$:      BRW     ACCVIO
                          06CD  1106
                          06CD  1107 ITM_PGFLCHAR:                            ; PAGE FILE CHARACTERISTICS
      58   AA    67    B0   06CD  1108          MOVW    (R7),PCB$W_PGFLCHAR(R10)
                          06D1  1109          RSB
```

```
                     06D2  1110
                     06D2  1111 ITM_PGFLINDEX:                                  ; PAGE FILE INDEX
5A AA   67   90      06D2  1112          MOVB    (R7),PCB$B_PGFLINDEX(R10)
             05      06D6  1113          RSB
                     06D7  1114
                     06D7  1115 ITM_INPUT_ATT:                                  ; SYS$INPUT ATTRIBUTES
78 A9   67   D0      06D7  1116          MOVL    (R7),PQB$L_INPUT_ATT(R9)
             05      06DB  1117          RSB
                     06DC  1118
                     06DC  1119 ITM_OUTPUT_ATT:                                 ; SYS$INPUT ATTRIBUTES
7C A9   67   D0      06DC  1120          MOVL    (R7),PQB$L_OUTPUT_ATT(R9)
             05      06E0  1121          RSB
                     06E1  1122
                     06E1  1123 ITM_ERROR_ATT:                                  ; SYS$INPUT ATTRIBUTES
0080 C9 67   D0      06E1  1124          MOVL    (R7),PQB$L_ERROR_ATT(R9)
             05      06E6  1125          RSB
                     06E7  1126
                     06E7  1127 NO_ITMLST:
```

```
                                06E7  1129          .SBTTL  ACTIVATE NEW PROCESS
                                06E7  1130 ;
                                06E7  1131 ;       ACTIVATE NEW PROCESS
                                06E7  1132 ;
                                06E7  1133
                                06E7  1134 ACTIVATE:                                    ;
            4C AA    59    D0   06E7  1135          MOVL    R9,PCB$L_PQB(R10)          ; POINT NEW PCB TO PROCESS QUOTA BLOCK
               54    5A    D0   06EB  1136          MOVL    R10,R4                     ; PCB ADDRESS OF NEW PROCESS
               6A    54    D0   06EE  1137          MOVL    R4,(R10)                   ; BUILD QUEUE HEADER
            04 AA    54    D0   06F1  1138          MOVL    R4,4(R10)                  ; FOR PCB
                                06F5  1139 10$:     DSBINT  W^10'01$                   ; BLOCK SYSTEM EVENT REPORTING
         00000000'GF    16    06FD  1140          JSB     G^MMG$CALCSWAPSIZE         ; GET THE SIZE IN R2 OF MINIMUM SWAP AREA
                  50    7C    0703  1141          CLRQ    R0                         ; INDICATE NO CURRENT SPACE
         00000000'GF    16    0705  1142          JSB     G^MMG$ALLOCSWPAREA         ; ALLOCATE INITIAL SWAP SPACE
                  08    18    070B  1143          BGEQ    15$                        ; IS SWAP SPACE AVAILABLE
         50    0000'8F    3C    070D  1144          MOVZWL  #SS$_INSSWAPSPACE,R0       ; NO
                  0105    31    0712  1145          BRW     65$
            20 A4    50    CE   0715  1146 15$:     MNEGL   R0,PCB$L_WSSWP(R4)         ; INSERT SWAP FILE TYPE VBN/INDEX
            5C A4    52    D0   0719  1147          MOVL    R2,PCB$L_SWAPSIZE(R4)      ; SAVE MAXIMUM ALLOWABLE SWAP SIZE
            56   FC AD    D0   071D  1148          MOVL    CURPCB(FP),R6              ; GET PCB ADDRESS OF CREATOR
            55   6C A6    D0   0721  1149          MOVL    PCB$L_PHD(R6),R5           ; AND EXTRACT HEADER ADDRESS
                                0725  1150 ;
                                0725  1151 ; Look for a free PCB slot (i.e. one pointing to nullpcb).  Start at the slot after
                                0725  1152 ; last PIX allocated and perform a round-robin scan.
                                0725  1153 ;
      53   00000000'EF    3C   0725  1154          MOVZWL  SCH$GL_MAXPIX,R3           ; SAVE MAX PIX TO TEST WHEN TO WRAP AROUND
         51    53    01'    C3   072C  1155          SUBL3   S^#<SCH$C_SWPPIX+1>,R3,R1  ; LOOP COUNTER IS MAX LESS SWAPPER AND NUL
      57   00000000'EF    D0   0730  1156          MOVL    SCH$GL_PIXLAST,R7          ; SET INDEX FOR PIX SEARCH TO LAST ALLOCATED
      58   00000000'EF    DE   0737  1157          MOVAL   SCH$GL_NULLPCB,R8          ; REFERENCE PCB ADDRESS (NULL PROCESS)
                  57    D6   073E  1158 20$:     INCL    R7                         ; MOVE TO THE NEXT PIX IN THE SEARCH
               53    57    D1   0740  1159          CMPL    R7, R3                     ; IS THE PIX LARGER THAN THE MAXIMUM
                  03    15   0743  1160          BLEQ    21$                        ; BRANCH IF R7 IS OK
               57    01'    D0   0745  1161          MOVL    S^#<SCH$C_SWPPIX+1>,R7     ; SET TO FIRST SLOT AFTER SWAPPER
      00000000'FF47    58    D1   0748  1162 21$:     CMPL    R8,@SCH$GL_PCBVEC[R7]      ; FIND NON-ZERO PIX POINTING TO NULLPCB
                  21    13   0750  1163          BEQL    30$                        ; GOT ONE, FREE SLOT
                  E9 51    F5   0752  1164          SOBGTR  R1,20$                     ; OCCUPIED, TRY ANOTHER
                                0755  1165 ;
                                0755  1166 ; Error, deallocate the paging file space
                                0755  1167 ;
      53    50    08    18    EF   0755  1168 22$:     EXTZV   #24,#8,R0,R3               ; GET PAGE FILE ALLOCATION IN
      50    50    18    00    EF   075A  1169          EXTZV   #0,#24,R0,R0               ; GET VBN OF ALLOCATION
                  51    52    D0   075F  1170          MOVL    R2,R1                      ; PASS SIZE OF SWAP SLOT TO DEALLOCATE
      53   00000000'FF43    D0   0762  1171          MOVL    @L^MMG$GL_PAGSWPVC[R3],R3  ; GET ADDR OF PAGE FILE CONTROL BLOCK
            00000000'GF    16    076A  1172          JSB     G^MMG$DEALLOCPAGFIL        ; FREE UP THE SPACE
                  00A2    31   0770  1173          BRW     60$                        ; NO FREE SLOTS AVAILABLE
                                0773  1174 ;
00000000'EF   00000000'EF    B1   0773  1175 30$:     CMPW    SCH$GW_PROCLIM,SCH$GW_PROCCNT ; CHECK FOR MAX PROCESSES
                  D5    15   077E  1176          BLEQ    22$                        ; BR IF YES AND ABORT CREATE
                                0780  1177 ;
                                0780  1178 ; Update global data and create the internal and external process identifiers.
                                0780  1179 ;
         00000000'EF    57    D0   0780  1180          MOVL    R7,SCH$GL_PIXLAST          ; SAVE NEW PIX AS LAST ALLOCATED PIX
            00000000'EF    B6   0787  1181          INCW    SCH$GW_PROCCNT             ; COUNT THIS PROCESS
                                078D  1182
                                078D  1183 ; Note that the following code assumes that caller's of Create Process
                                078D  1184 ; know what they are doing in the sense that there will never be more than
                                078D  1185 ; one of the bits INTER, BATCH, and NETWRK set at the same time.
```

```
                     078D  1186
                5B   D5  078D  1187          TSTL    R11
                16   13  078F  1188          BEQL    33$                     ; IT IS A SUBPROCESS - DON'T COUNT IT
          06 24 AA   19  E1  0791  1189          BBC     #PCB$V_INTER,PCB$L_STS(R10),31$
            00000000'EF  B6  0796  1190          INCW    SYS$GW_IJOBCNT          ; ONE MORE INTERACTIVE JOB
          06 24 AA   0E  E1  079C  1191 31$:     BBC     #PCB$V_BATCH,PCB$L_STS(R10),33$
            00000000'EF  B6  07A1  1192          INCW    SYS$GW_BJOBCNT          ; ONE MORE BATCH JOB
                     07A7  1193 33$:
                     07A7  1194 ;
                     07A7  1195 ; NOTE: The call to EXE$IPID_TO_EPID checks to make sure the IPID is valid,
                     07A7  1196 ;       therefore we must set the IPID in the new PCB and store the address
                     07A7  1197 ;       of the new PCB in the PCBVEC before we call EXE$IPID_TO_EPID.
                     07A7  1198 ;
  00000000'FF47   5A  D0  07A7  1199          MOVL    R10,@SCH$GL_PCBVEC[R7]  ; SET POINTER TO PCB IN VECTOR OF PCBS
50  00000000'FF47   3E  07AF  1200          MOVAW   @SCH$GL_SEQVEC[R7],R0   ; GET ADDRESS OF SEQUENCE NUMBER FOR SLOT
                60   B6  07B7  1201          INCW    (R0)                    ; NEXT SEQUENCE NUMBER FOR THIS PROCESS
                02   18  07B9  1202          BGEQ    35$                     ; BR IF IN RANGE (POSITIVE PID)
                60   B4  07BB  1203          CLRW    (R0)                    ; ELSE, RESET SEQUENCE NUMBER
             62 AA   60  B0  07BD  1204 35$:     MOVW    (R0),PCB$L_PID+2(R10)   ; SET SEQUENCE NUMBER
             60 AA   57  B0  07C1  1205          MOVW    R7,PCB$L_PID(R10)       ; AND PIX TO FORM COMPLETE INTERNAL PID
              50   60 AA  D0  07C5  1206          MOVL    PCB$L_PID(R10),R0       ; LOAD THE INTERNAL PID TO PASS TO ROUTINE
            00000000'EF  16  07C9  1207          JSB     EXE$IPID_TO_EPID        ; CONVERT IPID TO EPID, RETURN EPID IN R0
             64 AA   50  D0  07CF  1208          MOVL    R0,PCB$L_EPID(R10)      ; STORE THE EXTENDED PID
                     07D3  1209
                5B   D5  07D3  1210          TSTL    R11                     ; DETACHED CREATE?
                0A   13  07D5  1211          BEQL    38$                     ; BR IF NOT
           50   0080 CA  D0  07D7  1212          MOVL    PCB$L_JIB(R10),R0       ; GET JIB ADDRESS
         54 A0    60 AA  D0  07DC  1213          MOVL    PCB$L_PID(R10),JIB$L_MPID(R0)   ; AND SET ROOT PID FOR PROCESS TREE
                52   04  9A  07E1  1214 38$:     MOVZBL  #PRI$_TICOM,R2          ; SET PRIORITY INCREMENT CLASS
            00000000'EF  16  07E4  1215          JSB     SCH$CASE                ; MAKE PROCESS EXECUTABLE, NON-RESIDENT
                5B   D5  07EA  1216          TSTL    R11                     ; TEST FOR DETACHED CREATE
                0D   12  07EC  1217          BNEQ    40$                     ; BR IF CREATING DETACHED PROCESS
             42 A6   B6  07EE  1218          INCW    PCB$W_PRCCNT(R6)        ; OTHERWISE ACCOUNT FOR SUB-PROCESS
             5C A5   D5  07F1  1219          TSTL    PHD$L_CPULIM(R5)        ; CHECK FOR NO CPU LIMIT
                05   13  07F4  1220          BEQL    40$                     ; NO LIMIT, DONT DEDUCT
       5C A5    18 A9  C2  07F6  1221          SUBL    PQB$L_CPULM(R9),PHD$L_CPULIM(R5); DEDUCT CPU TIME LIMIT
                     07FB  1222 40$:                                         ;
          50   04 AE  D0  07FB  1223          MOVL    4(SP),R0                ; GET EPID RETURN VALUE ADDRESS
                0F   13  07FF  1224          BEQL    55$                     ; NONE
                1C   19  0801  1225          BLSS    70$                     ; SYSTEM SPACE ADDRESS
                     0803  1226 42$:     ENBINT                              ; RESTORE IPL
            60   64 AA  D0  0806  1227 45$:     MOVL    PCB$L_EPID(R10),(R0)    ; RETURN EXTENDED PID FOR CREATED PROCESS
          50   0000'8F  3C  080A  1228 50$:     MOVZWL  #SS$_NORMAL,R0          ; SET NORMAL COMPLETION STATUS
                04   080F  1229          RET                                 ; AND RETURN TO CALLER
                     0810  1230
                     0810  1231 55$:     ENBINT                              ; RESTORE IPL
                F5   11  0813  1232          BRB     50$                     ; AND EXIT
                     0815  1233
          50   0000'8F  3C  0815  1234 60$:     MOVZWL  #SS$_NOSLOT,R0          ; SET ERROR CODE FOR NO SLOT AVAILABLE
                     081A  1235 65$:     ENBINT                              ; ENABLE SYSTEM EVENT REPORTING
                21   11  081D  1236          BRB     ABORT                   ; ABORT CREATION, EXCEEDED QUOTA
                     081F  1237
  00000000'EF   50   D1  081F  1238 70$:     CMPL    R0,MMG$GL_NPAGEDYN      ; CHECK FOR IN NONPAGED POOL
                DB   1F  0826  1239          BLSSU   42$                     ; NO, DROP IPL TO STORE EPID
  00000000'EF   50   D1  0828  1240          CMPL    R0,EXE$GL_INTSTK        ; HIGH LIMIT
                D2   1E  082F  1241          BGEQU   42$                     ; OUT OF POOL OR INTERRUPT STACK
             60   64 AA  D0  0831  1242          MOVL    PCB$L_EPID(R10),(R0)    ; STORE EXTENDED PID FOR CREATED PROCESS
```

```
                     0835  1243          ENBINT
                     0838  1244          BRB     50$                       ; RESTORE IPL
                                                                           ; AND RETURN SUCCESS
                     083A  1245 10001$:                                    ; MARKER FOR END OF NONPAGABLE AREA
00000008             083A  1246          .LONG   IPL$_SYNCH                ; IPL TO BLOCK EVENT REPORTING
                     083E  1247 10002$:
                     083E  1248          ASSUME  <10002$-10$> LE 512       ; MAKE SURE IT IS LESS THAN A PAGE
```

DO 11

```
                                 083E  1250          .SBTTL  ABORT PROCESS CREATION
                                 083E  1251 ;
                                 083E  1252 ;       ABORT PROCESS CREATION AFTER DETECTING ANY ERROR
                                 083E  1253 ;
                                 083E  1254
                                 083E  1255
                                 083E  1256 ABRT2:                                          ; ABORT WITH PCB ONLY
                      59   D4    083E  1257          CLRL    R9                            ; INDICATE NO PQB ALLOCATED
                                 0840  1258 ABORT:                                          ; ABORT WITH BOTH PCB AND PQB ALLOCATED
                   57 50   D0    0840  1259          MOVL    R0,R7                         ; SAVE STATUS CODE
                      5B   D5    0843  1260          TSTL    R11                           ; IS THIS A DETACHED PROCESS CREATE?
                      0B   13    0845  1261          BEQL    10$                           ; BR IF NOT
               50 0080 CA   D0    0847  1262          MOVL    PCB$L_JIB(R10),R0             ; GET JIB ADDRESS FOR RELEASE
                      04   13    084C  1263          BEQL    10$                           ; BR IF NONE
                      33   10    084E  1264          BSBB    EXE_DEANONPAGED               ; DEALLOCATE JIB
                      14   11    0850  1265          BRB     20$                           ;
                 54   FC AD   D0    0852  1266 10$:     MOVL    CURPCB(FP),R4             ; GET PCB ADDRESS FOR SURE
               52 0080 C4   D0    0856  1267          MOVL    PCB$L_JIB(R4),R2              ; FETCH JIB ADDRESS
         3C A2  00000000'8F   C0    085B  1268          ADDL    #SWP$C_SHELLPFIL,JIB$L_PGFLCNT(R2); RETURN PAGE FILE QUOTA
                                 0863  1269                                                ; FOR SHELL PAGES
                   44 A2   B7    0863  1270          DECW    JIB$W_PRCCNT(R2)              ; CORRECT SUBPROCESS COUNT
                                 0866  1271
               50 00AC CA   D0    0866  1272 20$:     MOVL    PCB$Q_PRIV+ARB$L_RIGHTSLIST+8(R10),R0   ; GET EXTENDED RIGHTS LIST,
                      02   13    086B  1273          BEQL    22$                           ; BRANCH IF NONE
                      14   10    086D  1274          BSBB    EXE_DEANONPAGED               ; DEALLOCATE RIGHTS LIST
                   50 5A   D0    086F  1275 22$:     MOVL    R10,R0                        ; ADDRESS OF NEW PCB
                      0F   10    0872  1276          BSBB    EXE_DEANONPAGED               ; DEALLOCATE QUOTA BUFFER
                      59   D5    0874  1277          TSTL    R9                            ; ADDRESS OF PQB IF ANY
                      07   13    0876  1278          BEQL    30$                           ; BR IF NONE ALLOCATED
      00000000'FF     69   0E    0878  1279          INSQUE  (R9),@EXE$GL_PQBBL            ; DEALLOCATE PQB TO LOOKASIDE LIST
                   50 57   D0    087F  1280 30$:     MOVL    R7,R0                         ; RESTORE STATUS CODE
                      04   0882  1281          RET                                       ; AND RETURN TO CALLER
                                 0883  1282
                                 0883  1283 EXE_DEANONPAGED:
 00000000'GF     17              0883  1284          JMP     G^EXE$DEANONPAGED             ; BRANCH AID TO REACH THIS ROUTINE
```

```
                     0889  1286          .SBTTL  MOVSTR - STRING COPY SUBROUTINE
                     0889  1287 ;+
                     0889  1288 ; FUNCTIONAL DESCRIPTION:
                     0889  1289 ;       MOVSTR VALIDATES AND COPIES A STRING FROM THE ARGUMENT LIST
                     0889  1290 ;       TO THE PROCESS QUOTA BUFFER.  IF ANY ERROR IS DETECTED, THE
                     0889  1291 ;       SERVICE CALL IS EXITED VIA A RET INSTRUCTION WITH R0 CONTAINING
                     0889  1292 ;       THE ERROR STATUS CODE.
                     0889  1293 ;
                     0889  1294 ; CALLING SEQUENCE:
                     0889  1295 ;       BSB     MOVSTR
                     0889  1296 ;       .BYTE   <MAXIMUM_STRING_LENGTH>
                     0889  1297 ;       .BYTE   <AP_OFFSET_TO_SOURCE_DESCRIPTOR>
                     0889  1298 ;       .BYTE   <PQB_OFFSET_OF_DESTINATION>
                     0889  1299 ;
                     0889  1300 ; INPUT PARAMETERS:
                     0889  1301 ;       R9 - PQB BASE ADDRESS
                     0889  1302 ;       @(SP) - MAXIMUM STRING LENGTH
                     0889  1303 ;       @(SP)+1 - AP OFFSET TO SOURCE STRING DESCRIPTOR
                     0889  1304 ;       @(SP)+2 - PQB OFFSET FOR DESTINATION COUNTED STRING
                     0889  1305 ;
                     0889  1306 ; OUTPUT PARAMETERS:
                     0889  1307 ;       SPECIFIED AREA IN PQB RECEIVES SOURCE STRING
                     0889  1308 ;
                     0889  1309 ; COMPLETION CODES:
                     0889  1310 ;       SS$_ACCVIO  -  ACCESS VIOLATION FETCHING DESCRIPTOR OR STRING
                     0889  1311 ;       SS$_IVLOGNAM - INVALID LOGICAL NAME (COUNT OUT OF RANGE)
                     0889  1312 ;
                     0889  1313 ;-
                     0889  1314
                     0889  1315 MOVSTR:                                      ; MOVE STRING TO PQB
      55    6E   D0  0889  1316          MOVL    (SP),R5                    ; GET BASE OF PARAMETERS
      6E    04   C0  088C  1317          ADDL    #4,(SP)                    ; INCREMENT RETURN ADDRESS
      01F0  8F   BB  088F  1318          PUSHR   #^M<R4,R5,R6,R7,R8>        ; SAVE REGISTERS
      58    85   9A  0893  1319          MOVZBL  (R5)+,R8                   ; GET STRING LIMIT
      53    85   9A  0896  1320          MOVZBL  (R5)+,R3                   ; GET ARGLIST OFFSET
   52    6C43   D0  0899  1321          MOVL    (AP)[R3],R2                ; FETCH DESCRIPTOR ADDRESS
            2E   13  089D  1322          BEQL    MOVEXIT                    ; NONE, EXIT
                     089F  1323          IFNORD  #8,(R2),ACCVIO             ; MUST BE ABLE TO READ DESCRIPTOR
      56    62   7D  08A5  1324          MOVQ    (R2),R6                    ; FETCH DESCRIPTOR
            56   B5  08A8  1325          TSTW    R6                         ; CHECK FOR NULL COUNT
            21   13  08AA  1326          BEQL    MOVEXIT                    ; YES, NULL STRING
      58    56   B1  08AC  1327          CMPW    R6,R8                      ; CHECK UPPER LIMIT ON STRING
            08   1B  08AF  1328          BLEQU   10$                        ; BR IF WITHIN LIMIT
50    0000'8F    3C  08B1  1329          MOVZWL  #SS$_IVLOGNAM,R0           ; SET ERROR CODE
         FF87    31  08B6  1330          BRW     ABORT                      ; AND ABORT CREATE
                     08B9  1331
                     08B9  1332 10$:      IFNORD  R6,(R7),ACCVIO             ; CHECK ACCESSIBILITY
      53    65   3C  08BF  1333          MOVZWL  (R5),R3                    ; GET PQB OFFSET
   53    6943   9E  08C2  1334          MOVAB   (R9)[R3],R3                ; COMPUTE ADDRESS IN PQB
      83    56   90  08C6  1335          MOVB    R6,(R3)+                   ; SET COUNT FOR STRING
63    67    56   28  08C9  1336          MOVC3   R6,(R7),(R3)               ; COPY STRING TO BUFFER
                     08CD  1337 MOVEXIT:                                    ;
      01F0  8F   BA  08CD  1338          POPR    #^M<R4,R5,R6,R7,R8>        ; RESTORE REGISTERS
                  05  08D1  1339          RSB                                ; AND RETURN
                     08D2  1340
50    0000'8F    3C  08D2  1341 ACCVIO:  MOVZWL  #SS$_ACCVIO,R0             ; SET ERROR CODE
         FF66    31  08D7  1342          BRW     ABORT                      ;
```

```
                     08DA  1344          .SUBTITLE       ALLOCPQB - Allocate PQB from paged pool
                     08DA  1345 ;+
                     08DA  1346 ; Functional Description:
                     08DA  1347 ;
                     08DA  1348 ;       This subroutine merely allocates a PQB from paged pool. If the
                     08DA  1349 ;       allocation fails, the process may be put into a resource wait state
                     08DA  1350 ;       but only after the already allocated PCB is deallocated and IPL is
                     08DA  1351 ;       lowered to zero.
                     08DA  1352 ;
                     08DA  1353 ; Input Parameters:
                     08DA  1354 ;
                     08DA  1355 ;       R4  - Address of PCB of creating process
                     08DA  1356 ;       R10 - Address of chunk of nonpaged pool that will become PCB
                     08DA  1357 ;             of new process.
                     08DA  1358 ;
                     08DA  1359 ; Implicit Input:
                     08DA  1360 ;
                     08DA  1361 ;       Running at IPL$_ASTDEL as a result of successful PCB allocation
                     08DA  1362 ;
                     08DA  1363 ; Ouptut Parameters:
                     08DA  1364 ;
                     08DA  1365 ;       If allocation is successful
                     08DA  1366 ;
                     08DA  1367 ;             R2 - Address of PQB
                     08DA  1368 ;
                     08DA  1369 ;       If the allocation fails and the creator waits for resources,
                     08DA  1370 ;
                     08DA  1371 ;             the PCB pointed to by R10 is deallocated and the process
                     08DA  1372 ;             is put into a resource wait state. When paged pool becomes
                     08DA  1373 ;             available, the process resumes execution at the beginning
                     08DA  1374 ;             of the service.
                     08DA  1375 ;
                     08DA  1376 ;       If the allocation fails and resource wait is disabled,
                     08DA  1377 ;
                     08DA  1378 ;             a simple failure status (SS$_INSFMEM) is returned.
                     08DA  1379 ;
                     08DA  1380 ; Side Effects:
                     08DA  1381 ;
                     08DA  1382 ;       R0 through R3 are modified
                     08DA  1383 ;-
                     08DA  1384
                     08DA  1385 ALLOCPQB:                                       ; Allocate process quota block
51   08C8 8F    3C   08DA  1386          MOVZWL  #PQB$C_LENGTH,R1               ; Set structure size
        51    DD   08DF  1387          PUSHL   R1                             ; Save request size
  50    FA'AF    9E   08E1  1388          MOVAB   B^DEALLOCATE_PCB,R0            ; Store address of error action routine
 00000000'GF    16   08E5  1389          JSB     G^EXE$ALOPAGWAIT               ; Attempt to allocate packet
        51 8ED0   08EB  1390          POPL    R1                             ; Restore request size
     08 50    E9   08EE  1391          BLBC    R0,10$                         ; If low bit clear, no packet allocated
08 A2   51    B0   08F1  1392          MOVW    R1,PQB$W_SIZE(R2)              ; Insert size of allocated block
        0D    9B   08F5  1393          MOVZBW  #DYN$C_PQB,-                   ; Insert data structure type
      0A A2        08F7  1394                  PQB$B_TYPE(R2)                 ;  and clear adjacent byte
              05   08F9  1395 10$:     RSB                                    ;  and return
                   08FA  1396
                   08FA  1397 DEALLOCATE_PCB:
   50   5A    D0   08FA  1398          MOVL    R10,R0                         ; Get new PCB address
      FF83    30   08FD  1399          BSBW    EXE_DEANONPAGED                ; Give it back
              05   0900  1400          RSB                                    ; Return to error code in ALOPAGWAIT
```

```
0901  1401
0901  1402          .END
```

```
ABORT                   00000840 R      02
ABRT2                   0000083E R      02
ACCVIO                  000008D2 R      02
ACL$W_SIZE            = 00000008
ACTIVATE                000006E7 R      02
ALLOCPQB                000008DA R      02
ARB$C_LENGTH          = 00000078
ARB$L_RIGHTSLIST      = 00000020
ARB$R_LOCALRIGHTS     = 00000038
ARB$R_RIGHTSDESC      = 00000030
BASPRI                = 00000024
BUG$_KRPEMPTY           ********    X   02
CTL$GB_MSGMASK          ********    X   02
CTL$GL_KRPFL            ********    X   02
CTL$GL_PHD              ********    X   02
CTL$GL_UAF_FLAGS        ********    X   02
CTL$GT_CLINAME          ********    X   02
CURPCB                = FFFFFFFC
DEALLOCATE_PCB          000008FA R      02
DEFAULT_NAMES           0000001C R      02
DYN$C_PQB             = 0000000D
ERROR                 = 00000014
EXE$ALLOCBUF            ********    X   02
EXE$ALLOCJIB            ********    X   02
EXE$ALLOCPCB            ********    X   02
EXE$ALOPAGWAIT          ********    X   02
EXE$CREPRC              00000044 RG     02
EXE$DEANONPAGED         ********    X   02
EXE$GL_INTSTK           ********    X   02
EXE$GL_PQBBL            ********    X   02
EXE$GL_PQBFL            ********    X   02
EXE$GQ_SYSDISK          ********    X   02
EXE$IPID_TO_EPID        ********    X   02
EXE_DEANONPAGED         00000883 R      02
IMAGE                 = 00000008
INPUT                 = 0000000C
IPL$_SYNCH            = 00000008
IPL_SYNCH               0000044C R      02
ITEMLIST                00000685 R      02
ITMLST                = 00000034
ITMLST_ARG            = 0000000D
ITM_ERROR_ATT           000006E1 R      02
ITM_INPUT_ATT           000006D7 R      02
ITM_OUTPUT_ATT          000006DC R      02
ITM_PGFLCHAR            000006CD R      02
ITM_PGFLINDEX           000006D2 R      02
JIB$C_LENGTH          = 00000074
JIB$L_BYTCNT          = 00000020
JIB$L_BYTLM           = 00000024
JIB$L_MPID            = 00000054
JIB$L_MTLBL           = 00000004
JIB$L_MTLFL           = C0000000
JIB$L_ORG_BYTLM       = 0000006C
JIB$L_PGFLCNT         = 0000003C
JIB$L_PGFLQUOTA       = 00000038
JIB$S_ACCOUNT         = 00000008
JIB$S_USERNAME        = 0000000C
JIB$T_ACCOUNT         = 00000018
JIB$T_USERNAME        = 0000000C
JIB$W_ENQCNT          = 0000004C
JIB$W_ENQLM           = 0000004E
JIB$W_FILCNT          = 00000030
JIB$W_FILLM           = 00000032
JIB$W_MAXDETACH       = 00000052
JIB$W_MAXJOBS         = 00000050
JIB$W_PRCCNT          = 00000044
JIB$W_PRCLIM          = 00000046
JIB$W_TQCNT           = 00000034
JIB$W_TQLM            = 00000036
LNM$SEARCH_ONE          ********    X   02
LNMX$T_XLATION        = 00000004
LNM_ATTR              = 00000103
LNM_TBL                 00000030 R      02
MAXQSCNT                0000064C R      02
MBXUNT                = 0000002C
MMG$ALLOCSWPAREA        ********    X   02
MMG$CALCSWAPSIZE        ********    X   02
MMG$DEALLOCPAGFIL       ********    X   02
MMG$GL_NPAGEDYN         ********    X   02
MMG$GL_PAGSWPVC         ********    X   02
MOVEXIT                 000008CD R      02
MOVSTR                  00000889 R      02
NOQLIST                 000004A0 R      02
NO_ITMLST               000006E7 R      02
OUTPUT                = 00000010
OVERCHECK               00000678 R      02
PCB$B_ASTEN           = 0000000D
PCB$B_PGFLINDEX       = 0000005A
PCB$B_PRI             = 0000000B
PCB$B_PRIB            = 0000002F
PCB$C_LENGTH          = 00000120
PCB$L_ARB             = 0000008C
PCB$L_ASTQBL          = 00000014
PCB$L_ASTQFL          = 00000010
PCB$L_DEFPROT         = 00000114
PCB$L_EOWNER          = 00000068
PCB$L_EPID            = 00000064
PCB$L_JIB             = 00000080
PCB$L_LOCKQBL         = 00000108
PCB$L_LOCKQFL         = 00000104
PCB$L_OWNER           = 0000001C
PCB$L_PHD             = 0000006C
PCB$L_PID             = 00000060
PCB$L_PQB             = 0000004C
PCB$L_STS             = 00000024
PCB$L_SWAPSIZE        = 0000005C
PCB$L_UIC             = 000000BC
PCB$L_WSSWP           = 00000020
PCB$M_SECAUDIT        = 08000000
PCB$Q_PRIV            = 00000084
PCB$T_LNAME           = 00000070
PCB$V_BATCH           = 0000000E
PCB$V_DISAWS          = 00000018
PCB$V_HIBER           = 00000013
```

Symbol table

```
PCB$V_INTER                     = 00000019
PCB$V_LOGIN                     = 00000014
PCB$V_NETWRK                    = 00000015
PCB$V_NOACNT                    = 0000000F
PCB$V_PSWAPM                    = 00000004
PCB$V_SSFEXCU                   = 00000009
PCB$V_SSRWAIT                   = 0000000A
PCB$W_ASTCNT                    = 00000038
PCB$W_BIOCNT                    = 0000003A
PCB$W_BIOLM                     = 0000003C
PCB$W_DIOCNT                    = 0000003E
PCB$W_DIOLM                     = 00000040
PCB$W_GRP                       = 000000BE
PCB$W_PGFLCHAR                  = 00000058
PCB$W_PPGCNT                    = 00000036
PCB$W_PRCCNT                    = 00000042
PCB$W_TMBU                      = 00000032
PCB_IB                          = 02004000
PCB_IBN                         = 02204000
PHD$L_CPULIM                    = 0000005C
PHD$L_CPUTIM                    = 00000038
PHD$R_MAX_CLASS                 = 00000128
PHD$R_MIN_CLASS                 = 00000114
PHD$S_MAX_CLASS                 = 00000014
PHD$S_MIN_CLASS                 = 00000014
PHD$W_ASTLM                     = 00000040
PHD$W_WSEXTENT                  = 00000016
PHD$W_WSLIST                    = 00000008
PHD$W_WSQUOTA                   = 00000018
PIDADR                          = 00000004
PIO$GT_DDSTRING                   ********    X   02
PQB$B_MSGMASK                   = 00000046
PQB$B_TYPE                      = 0000000A
PQB$C_LENGTH                    = 000008C8
PQB$L_ASTLM                     = 0000000C
PQB$L_CPULM                     = 00000018
PQB$L_CREPRC_FLAGS              = 0000004C
PQB$L_ERROR_ATT                 = 00000080
PQB$L_INPUT_ATT                 = 00000078
PQB$L_OUTPUT_ATT                = 0000007C
PQB$L_UAF_FLAGS                 = 00000048
PQB$L_WSQUOTA                   = 00000030
PQB$M_IMGDMP                    = 00000001
PQB$Q_PRVMSK                    = 00000000
PQB$R_MAX_CLASS                 = 00000064
PQB$R_MIN_CLASS                 = 00000050
PQB$S_CLI_NAME                  = 00000020
PQB$S_CLI_TABLE                 = 00000100
PQB$S_DDSTRING                  = 00000100
PQB$S_MAX_CLASS                 = 00000014
PQB$S_MIN_CLASS                 = 00000014
PQB$S_SPAWN_CLI                 = 00000020
PQB$S_SPAWN_TABLE               = 00000100
PQB$T_CLI_NAME                  = 00000088
PQB$T_CLI_TABLE                 = 000000A8
PQB$T_DDSTRING                  = 000006C8
PQB$T_DISK                      = 000005C8
PQB$T_ERROR                     = 000004C8
PQB$T_IMAGE                     = 000007C8
PQB$T_INPUT                     = 000002C8
PQB$T_OUTPUT                    = 000003C8
PQB$T_SPAWN_CLI                 = 000001A8
PQB$T_SPAWN_TABLE               = 000001C8
PQB$W_FLAGS                     = 00000044
PQB$W_SIZE                      = 00000008
PQL$AC_DEFAULT                    ********    X   02
PQL$AL_MIN                        ********    X   02
PQL$_LENGTH                     = 0000000F
PQL$_LISTEND                    = 00000000
PQL_MDEDUCT                     = 00000000
PR$_IPL                           ********    X   02
PRC$V_BATCH                     = 00000004
PRC$V_CLISPEC                   = 0000000C
PRC$V_DETACH                    = 00000009
PRC$V_DISAWS                    = 00000008
PRC$V_HIBER                     = 00000005
PRC$V_IMGDMP                    = 0000000B
PRC$V_INTER                     = 0000000A
PRC$V_NETWRK                    = 00000007
PRC$V_NOACNT                    = 00000003
PRC$V_NOPASSWORD                = 0000000D
PRC$V_NOUAF                     = 00000006
PRC$V_PSWAPM                    = 00000002
PRC$V_SSFEXCU                   = 00000001
PRC$V_SSRWAIT                   = 00000000
PRC$_ERROR_ATT                  = 00000005
PRC$_INPUT_ATT                  = 00000003
PRC$_OUTPUT_ATT                 = 00000004
PRC$_PGFLCHAR                   = 00000001
PRC$_PGFLINDEX                  = 00000002
PRCNAM                          = 00000020
PRI$_TICOM                      = 00000004
PRV$V_CMKRNL                    = 00000000
PRV$V_DETACH                    = 00000005
PRV$V_NOACNT                    = 00000009
PRV$V_PSWAPM                    = 0000000C
PRV$V_SETPRI                    = 0000000D
PRV$V_SETPRV                    = 0000000E
PRVADR                          = 00000018
QASTLM                            000004FE  R       02
QBIOLM                            00000511  R       02
QBYTLM                            00000528  R       02
QCPULM                            00000547  R       02
QDIOLM                            00000580  R       02
QENQLM                            0000065D  R       02
QFILLM                            00000597  R       02
QJTQUOTA                          000004FD  R       02
QPGFLQUOTA                        000005B2  R       02
QPRCLM                            000005CD  R       02
QTQELM                            000005E4  R       02
QUOTA                           = 0000001C
QUOTALIST                         00000450  R       02
QWSDEFAULT                        0000063F  R       02
QWSEXTENT                         0000061A  R       02
```

```
QWSQUOTA                        000005FF R      02
SCH$CHSE                        ********   X    02
SCH$C_SWPPIX                    ********   X    02
SCH$GL_MAXPIX                   ********   X    02
SCH$GL_NULLPCB                  ********   X    02
SCH$GL_PCBVEC                   ********   X    02
SCH$GL_PIXLAST                  ********   X    02
SCH$GL_SEQVEC                   ********   X    02
SCH$GL_SWPPID                   ********   X    02
SCH$GW_PROCCNT                  ********   X    02
SCH$GW_PROCLIM                  ********   X    02
SGN$GL_MAXWSCNT                 ********   X    02
SS$_ACCVIO                      ********   X    02
SS$_BADPARAM                    ********   X    02
SS$_DUPLNAM                     ********   X    02
SS$_EXPRCLM                     ********   X    02
SS$_EXQUOTA                     ********   X    02
SS$_INSSWAPSPACE                ********   X    02
SS$_IVLOGNAM                    ********   X    02
SS$_IVQUOTAL                    ********   X    02
SS$_IVSTSFLG                    ********   X    02
SS$_NOLOGNAM                    ********   X    02
SS$_NOPRIV                      ********   X    02
SS$_NORMAL                      ********   X    02
SS$_NOSLOT                      ********   X    02
STSFLG                        = 00000030
STSFLGCNT                     = 0000000E
STSFLGTBL                       00000000 R      02
SWP$C_SHELLPFIL                 ********   X    02
SWP$GL_SHELLSIZ                 ********   X    02
SYS$GW_BJOBCNT                  ********   X    02
SYS$GW_FILEPROT                 ********   X    02
SYS$GW_IJOBCNT                  ********   X    02
UIC                           = 00000028
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
ABS	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
Y$EXEPAGED	00000901 (2305.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.06	00:00:01.94
Command processing	114	00:00:00.59	00:00:04.91
Pass 1	352	00:00:12.96	00:00:51.86
Symbol table sort	0	00:00:01.50	00:00:05.40
Pass 2	265	00:00:03.89	00:00:15.75
Symbol table output	31	00:00:00.22	00:00:00.45

```
Psect synopsis output              2     00:00:00.02    00:00:00.02
Cross-reference output             0     00:00:00.00    00:00:00.00
Assembler run totals             795     00:00:19.25    00:01:20.65

The working set limit was 1650 pages.
74594 bytes (146 pages) of virtual memory were used to buffer the intermediate code.
There were 60 pages of symbol table space allocated to hold 969 non-local and 111 local symbols.
1402 source lines were read in Pass 1, producing 22 object records in Pass 2.
33 pages of virtual memory were used to define 32 macros.
```

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SYS.OBJ]LIB.MLB;1	18
_255DUA28:[SYSLIB]STARLET.MLB;2	9
TOTALS (all libraries)	27

```
1034 GETS were required to define 27 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:SYSCREPRC/OBJ=OBJ$:SYSCREPRC MSRC$:SYSCREPRC/UPDATE=(ENH$:SYSCREPRC)+EXECML$/LIB
```

SYSCANEVT
LIS

SYSCHKPRO
LIS

SYSCREDEL
LIS

SYSCREPRC
LIS

SYSCANCEL
LIS

SYSCOMMON
LIS

SYSCHGMOD
LIS